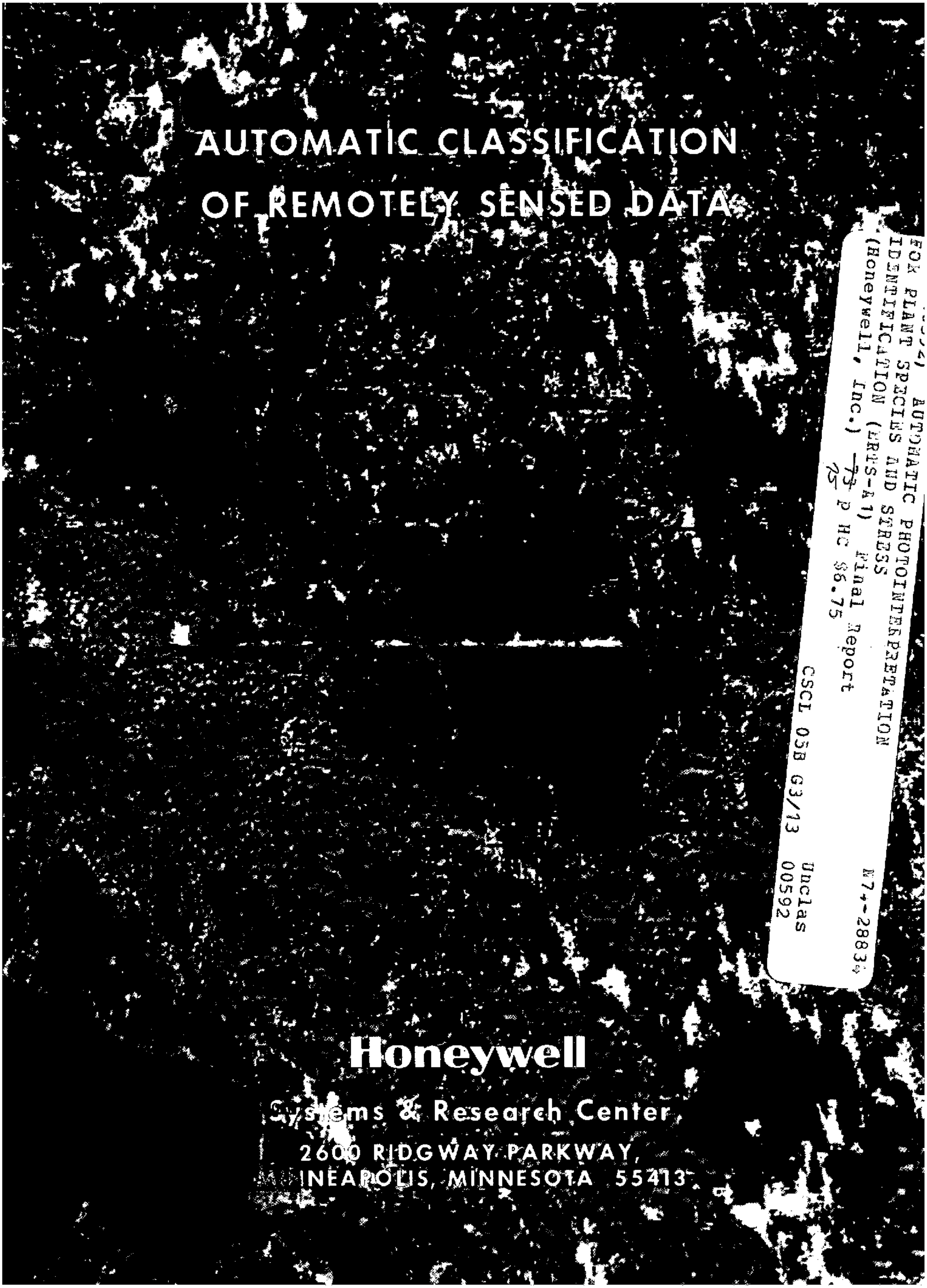


AUTOMATIC CLASSIFICATION OF REMOTELY SENSED DATA

Honeywell

Systems & Research Center

2600 RIDGWAY PARKWAY,
MINNEAPOLIS, MINNESOTA 55413

Cover Photo:

Thematic map of Chippewa National Forest in Northern Minnesota made from an October and January ERTS Coverage.

CLASS	COLOR
Open	White
Cutover	Yellow
Marsh	Yellow Green
Water	Blue
Mixed	Red
Conifer Lowland Low Density	Orange
Conifer Lowland High Density	Light Orange
Conifer Highland High Density	Brown Green
Hardwood Lowland Low Density	Dark Blue
Hardwood Lowland High Density	Purple
Hardwood Highland Low Density	Wine
Hardwood Highland High Density	Black

STIF

08

III

E7.4-10592
CR-138701

12 December 1973

FINAL REPORT
ON
AUTOMATIC PHOTOINTERPRETATION FOR PLANT SPECIES AND STRESS IDENTIFICATION (ERTS-A1)

ERTS
Proposal Number MMC 647
Principal Investigator Number PR 202

Contract Number NAS5-21866

Prepared by: L. Kirvida

Approved by: G.D.Swanlund

Honeywell, Inc.
Systems and Research Division
2600 Ridgway Pkwy.
Minneapolis, Minnesota 55413

1647A

AUTOMATIC PHOTOINTERPRETATION FOR PLANT SPECIES AND STRESS IDENTIFICATION

TABLE OF CONTENTS

Page

ABSTRACT

The objectives of this study were to determine if forest types could be delineated and whether stress could be detected using automatic techniques. We have shown that automatic photointerpretation is feasible for delineating forest species; there was not enough plant stress in the area to evaluate this aspect. We have also evaluated the use of texture for improving classification accuracy. In general, the classification accuracy with 5 or 6 classes ranges from 70 to 90%. Since chance performance is 16 to 20%, one can conclude that automatic photointerpretation is feasible. These performance figures are typical for distinguishing forest types to conifers and hardwoods. Using density alone for features, the classification accuracy for 5 classes is 70 to 75%. Adding texture improves this accuracy about 10 to 15%. Finer distinctions introduced higher errors, for example breaking conifers into highland vs. lowland and high density vs. low density results in an accuracy for these four classes of 69.7%. With four classes, chance would result in an accuracy of 25%, thus the improvement is not as striking.

FINAL REPORT

AUTOMATIC PHOTOINTERPRETATION FOR PLANT SPECIES AND STRESS IDENTIFICATION

Contract NAS5-21866

MMC 647

Introduction

The objective of this program is to evaluate the feasibility of using automatic interpretation techniques to map plant species and plant stress. In particular, Forest type mapping in Northern Minnesota has been selected as the problem to be addressed. We have worked closely with the School of Forestry at the University of Minnesota who has supplied us with ground truth in the area.

In this program we have used bulk ERTS photos for identifying the study areas and for preliminary evaluation of the data quality. However, computerized photo interpretation has been performed using the digital data only. The features generated from the data have been both multispectral and spatial. An evaluation of the usefulness of both types of features has been made. In addition, two seasonal coverages have been used to construct the feature vector and the performance achieved from multi-temporal coverage is compared with a single coverage.

Two test sites have been selected for this study based on the first available cloud-free coverage, adequate size and type of classes, and availability of ground truth. The University of Minnesota School of Forestry test site at Cloquet was selected as the pilot site for evaluating the spectral and texture algorithms. Later, the Chippewa National Forest was analyzed using multispectral features only in an effort to deliniate a larger area which is also a separate management unit. It is hoped that the availability of a thematic map of this area will serve to indicate the usefulness of remote sensing as an inventory and change detection tool and promote user.involvement. To this end, several meetings have been held with personnel from the Regional Forest Center of Milwaukee, the Superior National Forest and Chippewa National Forest.

The performance of the automatic classifier ranges from 70% to 99% depending on the number of classes; the number and complexity of the features; and finally, the similarity between the training and testing samples. The results of the automatic classifier are available in two forms. First, a confusion matrix is produced during the training procedures which shows the correct classifications and misclassifications for all samples in the training set. Secondly, a thematic map is generated for an area, which must be similar to the training area, based on the set of weights obtained in the training procedure. This delineation map can be visually compared with the ground truth map to obtain an evaluation of performance.

A comparison between the performance we achieved with the automatic classifier can be made with manual photointerpretation performed by experienced photointerpreters at the University of Minnesota School of Forestry, Institute of Agriculture Remote Sensing Laboratory. Both studies used portions of the Chippewa National Forest but not identical sites. Automatic classification was performed on six classes. A classification accuracy of 72.7% was achieved testing on the training set. Seven classes were manually classified by two photointerpreters with an accuracy of 45% and 41% using a color combination of bands 5 and 7 from the October coverage. However, it should be noted that the manual photointerpretation performance was based on the complete data set whereas the automatic classification performance was based on the training data only, i.e., the data contained no mixture of classes and a good representation of the various classes.

Experiment Design

Several candidate test sites were considered at the start of this study. These included the Superior National Forest, Chippewa National Forest, Itasca County, Koochiching County and Carlton County. All of these areas are heavily wooded.

Cloquet Test Site:

One of the first cloud free images received covered Carlton County (1075-16312 October 6, 1972). Since the Cloquet Forest test site is in the area, we decided to use this image for our pilot studies. The Cloquet Forest site is located about 25 miles west of Duluth. Ground truth information was provided by Greg Johnson from the University of Minnesota Institute of Agriculture Remote Sensing Laboratory located in the College of Forestry. Since the College's Cloquet Forestry Center, an experimental forest, is in the midst of this area, much information was previously known about the forest types. Spring 1:90,000 panchromatic aerial photographs, numerous field checks, and previous ground experience in the study area were used by the interpreters in generating the ground truth map. The Cloquet area was delineated into five types: conifers, hardwoods, open, water and city. Five thousand acres, approximately equally divided into the five classes, were used for training. We then delineated a total of 24,000 acres based on the training results.

The relatively small Cloquet area was used to evaluate multispectral and spatial features for automatic classification. The principal components technique was used to determine the most effective bands for class separation. In addition, four texture algorithms were evaluated for distinguishing between these five classes.

Data Analysis Procedures

Data from the bulk, black and white 70 mm transparencies and 7 track 800 BPI computer compatible tapes (CCT) were used as the data base. The imagery was used for orientation and registration, and the digital data was used to perform the automatic stratification and analysis.

The ERTS digital data of the study area was then reproduced on film by writing with a digital magnetic tape to film printer for purposes of registering with ground truth information. The film output for Band 7 is shown in Figure 1. It provides an image of the study area containing grid lines corresponding to record and word on the digital magnetic data tape. Registration of ground truth with ERTS-A data was accomplished by recognition of landmarks such as water bodies in the area. Registration with ground truth maps was required for both training and evaluating the automatic classification system and for producing the stratification output.

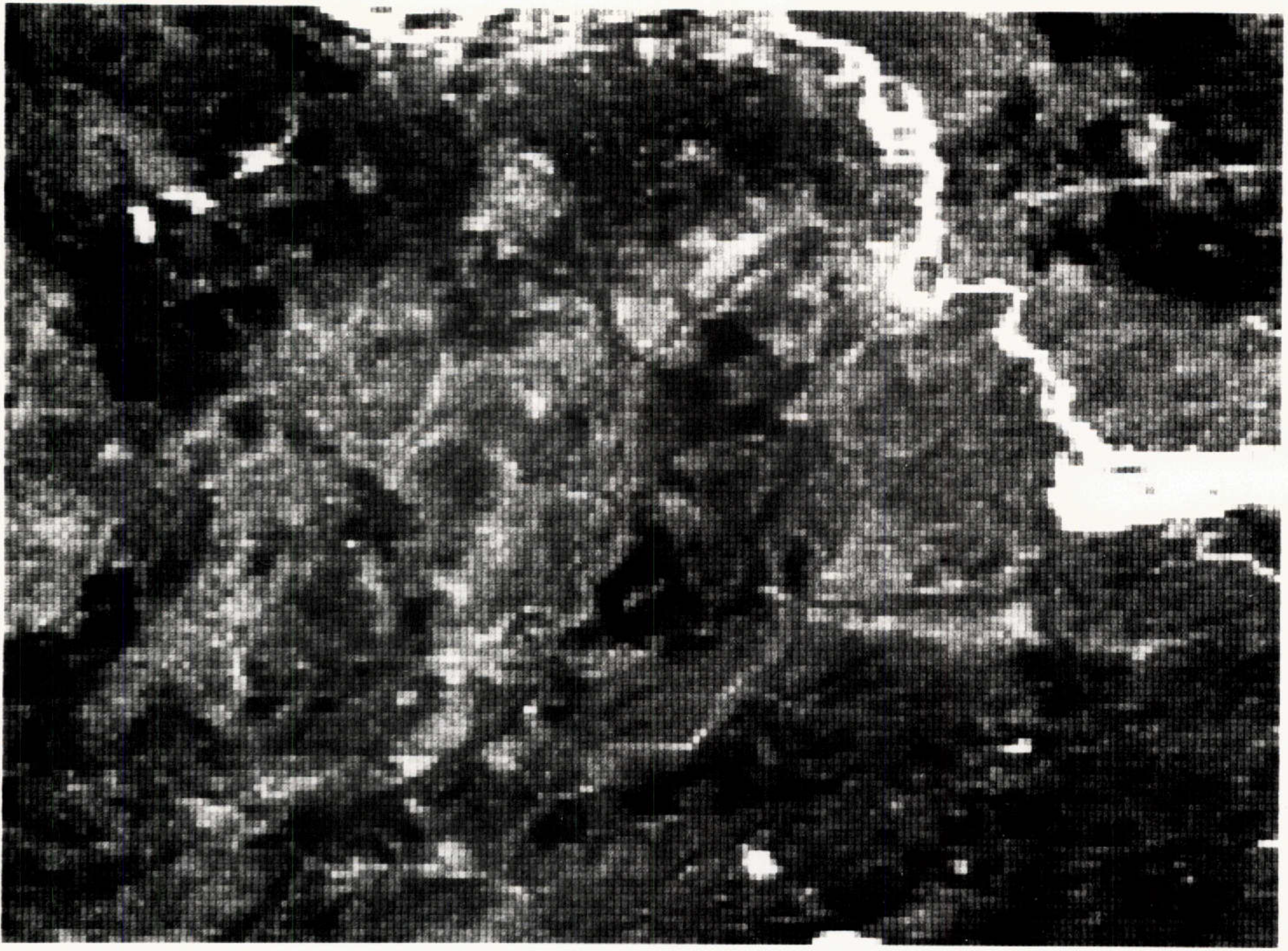
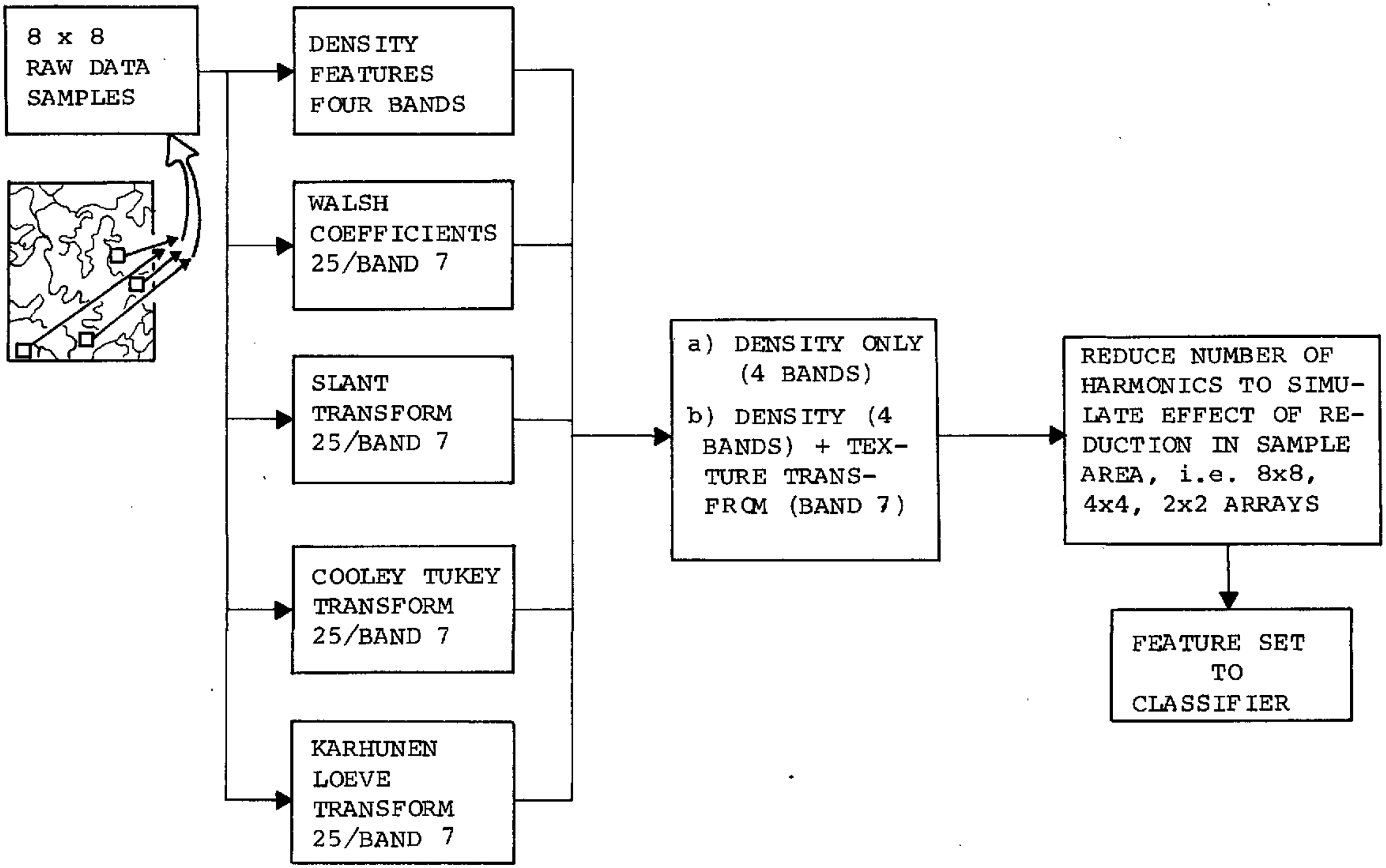


FIGURE I. BAND 7, CLOQUET TEST SITE
1075 - 16312 p. 4

Figure 2. ERTS-A Feature Extraction Procedure



Once ground truth and ERTS-A data were registered, type boundaries were encoded in terms of record and word numbers. From within the type boundaries, 8 x 8 arrays were isolated to serve as training samples. During training two categories of features, multispectral and texture, were generated for a number of 8 x 8 array samples in each class as illustrated in Figure 2.

The first experiment was run using only spectral features. An illustration of the separability between the five classes is shown by the density histograms in Figure 3. The four sets of histograms were derived from each of the four MSS bands. Band 4 has a great deal of overlap between classes, a part of which is due to the banding visible in Figure 1. Band 7 is excellent for separating water from land and was usually used for locating lakes for ground truth landmarks.

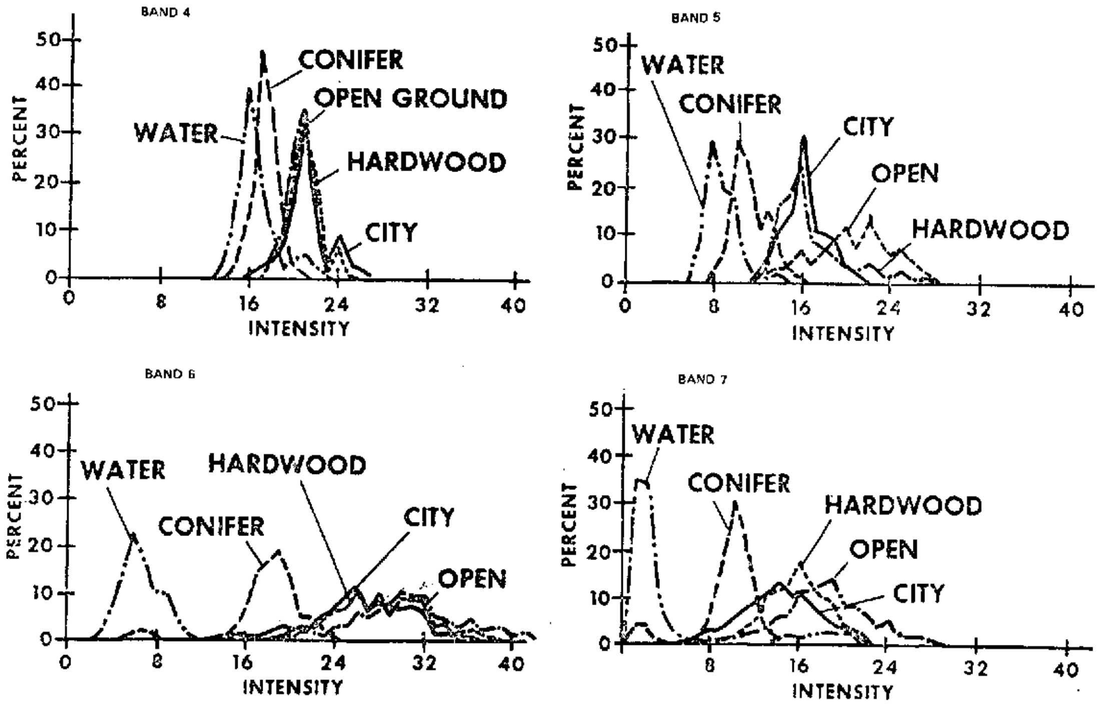


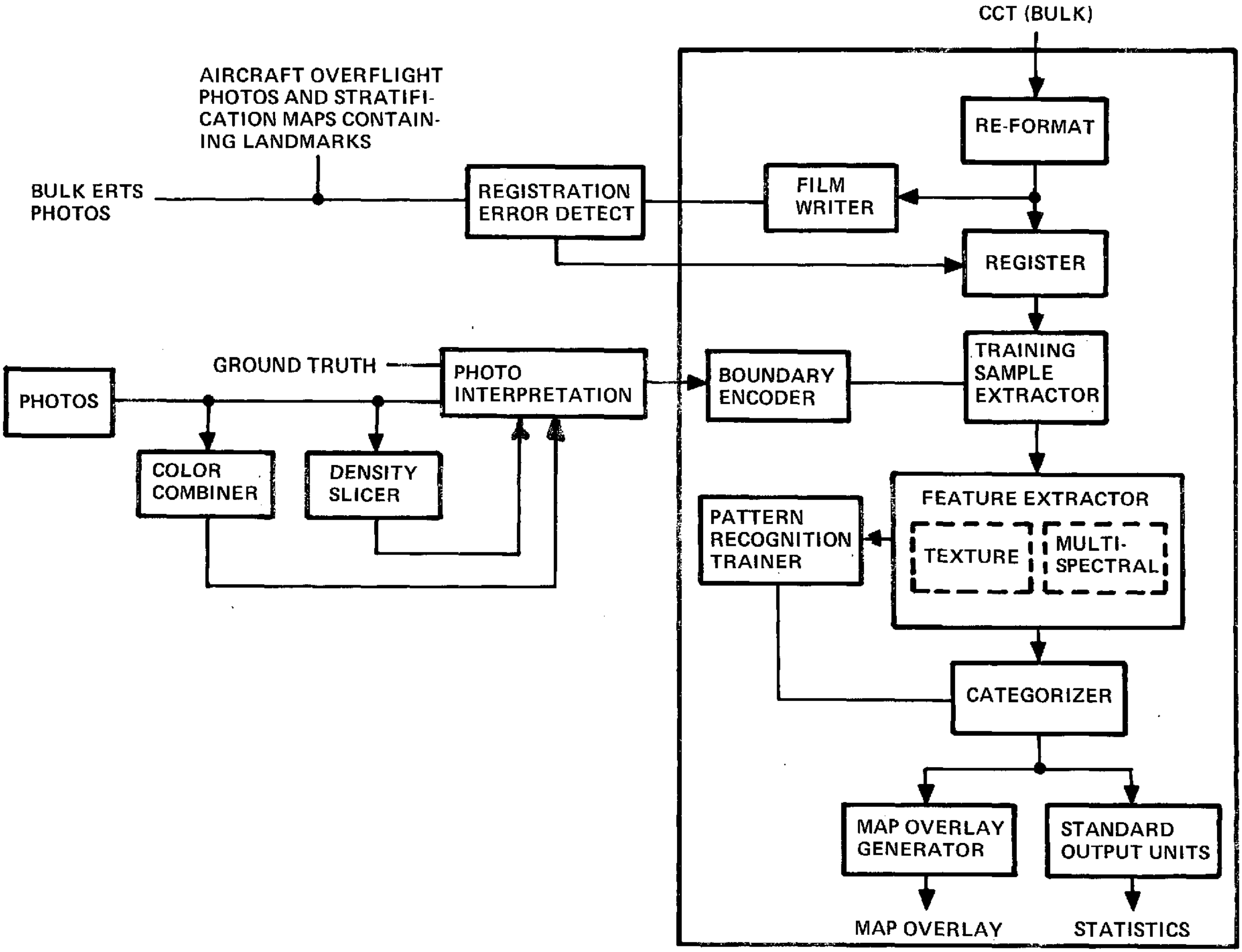
Figure 3. Histograms of Intensity Levels
ERTS 1075-16312

Two dimensional histograms forthe five classes were also computed. These indicated high cross correlation between some pairs of bands for a few classes; however, no consistent conclusions could be drawn.

For texture features, two dimensional Fast Walsh, Fast Fourier, Slant and Karhunen-Loeve Transforms were utilized. Texture features were computed from an 8 x 8 array representing approximately 70 ground acres. Texture was also computed on a 4 x 4 array to determine the effect of array size on performance. Increasing the array size increases frequency resolution which increases the classification accuracy. However, the larger array size also increases the minimum ground resolution area which can be classified.

Having selected the features to be used and the training set, a linear discriminant classifier is trained. Briefly, the classifier algorithm groups each of the features of the training set around an orthogonal basis vector in a least mean square sense. The "weight" matrix required to do this is computed for subsequent application to the input data during testing and during the generation of overlay maps. The class to which the input data point belongs is determined by the distance from the various orthogonal vector points. A block diagram illustrating the procedures used for automatic interpretation is shown in Figure 4.

FIGURE 4. **DATA ANALYSIS PROCEDURES**



Feature Selection

Multispectral and spatial features were used for automatic classification. The multispectral data consisted of the output of the four MSS sensors which are sensitive to .5-.6, .6-.7, .7-.8 and .8 to 1.1 microns. In addition, an evaluation of five spatial frequency algorithms was made on the Cloquet test set. These algorithms were the Fourier, Karhunen Loeve, Walsh and Slant transforms.

The spatial features were added in order to include pattern information. Edges in a picture introduce spatial frequencies along a line in the complex frequency plan orthogonal to the edge. High spatial frequencies correspond to sharp edges and low spatial frequencies correspond to regions of approximately uniform grey band. Spatial filtering in an image to detect texture is a natural extension to two dimensions of the traditional one-dimensional or temporal filtering process in communication networks.

The Fourier transform, which has been a commonly accepted tool for computing the frequency components of a temporal waveform, utilizes sinusoidal orthogonal basis functions. Digital implementation of the Fourier transform became feasible for two dimensions with the development by Cooley and Tukey of the Fast Fourier transform (FFT). The FFT was our first algorithm used to generate spatial features. Although it is inferior to the Karhunen Loeve transform in a mean square error sense, i.e., the first M coefficients of the Karhunen Loeve transform represent the data more accurately than in coefficients of the FFT, the FFT can be computed far more efficiently with $N^2\log_2 N$ computer operations where N is the dimensionality of the pattern space. A block diagram of the computations involved in this algorithm is given in the appendix.

The second algorithm used to measure spatial frequency was the Walsh Hadamard transform. This transformation has a number of advantages; it can be derived with $N^2\log_2 N$ additions or subtractions and is binary so that it is amenable to digital computation. Sequency is proportional to the number of zero crossings of the Walsh wave analogous to the sinusoidal frequency descriptor. The development of this algorithm is given in the appendix .

The third algorithm used for spatial features is the Slant transform. Pratt, et al., from the University of Southern California developed a computationally fast Slant algorithm. One of the advantages of the Slant transform is the compaction of the image energy into a minimum number of basis vectors which resemble typical horizontal or vertical lines of an image. Generally lines in an image will have a constant grey level over considerable length, or linearly vary in brightness over the length. The orthogonal set of basis functions in the Slant transform tend to accommodate this type of data. It also has a sequency property descriptive of frequency content. In fact, some of the basis vectors of the Slant transform are identical to the Walsh basis vectors. Pratt has shown that the mean square error between an image and the Slant transform is almost as small as that of the Karhunen Loeve transform. A further description of the algorithm is given in the appendix.

The fourth algorithm applied to the Cloquet test site data was the Karhunen Loeve expansion. For any data set, this algorithm minimizes the dimensionality of the feature space for a given truncation error. This algorithm is a linear transformation described by an orthogonal matrix. Such a transformation is equivalent to a rotation of the original pattern space to a new set of coordinate vectors which are also orthogonal and which have a number of advantages including a reduction of dimensionality. The disadvantages of this rotational transformation lie in its computational requirements. The covariance matrix must be computed and diagonalized. Finally, the pattern space must be rotated. Because of the relative difficulty of applying this transformation to a large data set and large number of classes, this transformation was only used as a comparative yardstick for measuring the performance of the previous three transforms.

The major portion of this study was directed at automatic classification using a discrimination rule, which is trained on a data set. This requires that the ground truth training set be reasonably accurate and repeatable and furthermore that registration between the ERTS digital data and ground truth be accurate, preferably to one pixel.

The constraint on registration accuracy is due to the requirement for correct training data. This requirement is basic to obtaining a correct set of training weights. The boundary between classes is determined by grouping samples of the training classes around the points of a simplex in a least mean square sense. If there are errors in the training class membership, these boundaries are going to be erroneously distorted and automatic stratification will likewise be in error.

To allow for misalignment between the ERTS data and ground truth, the training set samples were extracted from the central portions of the delineated area. Proximity to the boundaries between classes was avoided to allow for minor misalignment in the registration process.

To assure training on accurate ground truth, a trained photointerpreter, Mr. Greg Johnson, from the University of Minnesota Graduate School in Forestry delineated all ground truth. Recent aerial photos were utilized and ground checks were made for obtaining an up-to-date ground truth map. An attempt was made to obtain approximately equal areas of the various classes for training and testing since an overwhelming predominance of any of the classes would distort the statistical significance of the final decision. However, the automatic classifier decision rule can accomodate for inequalities in expected probabilities of the various classes.

Automatic Classification Algorithm

The first automatic classification procedure used in this study is called K-Class. This algorithm was developed at Honeywell. The theory of this algorithm and optimization procedures are described below. The program has been coded for use on the CDC 6600, the XDS 9300 and the DDP 24.

On each computer, the software is broken up into two programs. The first program (DISPERSION) computes the statistics used in the second program (KCLASS). On the CDC 6600, which has adequate memory, these two programs could be combined.

The K-Class algorithm for recognizing classes is a linear mapping from a measured feature space to a decision space.

That is

$$D = B^T X \qquad (1)$$

where

$D = (d_1, d_2, \ldots, d_K)^T$ is K x 1 decision vector

B = the linear mapping transformation (coefficient matrix)

$X = (X^1, X^2, \ldots, X^N, -1)^T$ is a (N+1) x 1 feature vector

The classification procedure is to classify as class i if

$$d_i > d_j \text{ for all } j \neq i \qquad (2)$$

A block diagram illustrating the computations envolved in K-class is shown in Figure 5. The measurements or attribute inputs are those characteristics of the input signal from which features are generated. These are the output signals from the four MSS bands for the various seasons. These measurements are used directly for features. In addition, texture features are computed from arrays of these measurements in the feature vector processor.

All of the feature vector samples cannot be mapped into the same point with one linear mapping since the feature space is statistical in nature. Thus, the decision space is also statistical. To find B we then minimize a mean square mapping error. For class j

$$e_j = E[||B^T X_j - A_j||] \qquad (3)$$

is the mean square mapping error, where A_j is the j^{th} column of an orthogonal decision space A. This is orthogonal so as to make the classes mathematically independent.

Because some classes have more error than others, we choose the total error to be a weighted mean square mapping error

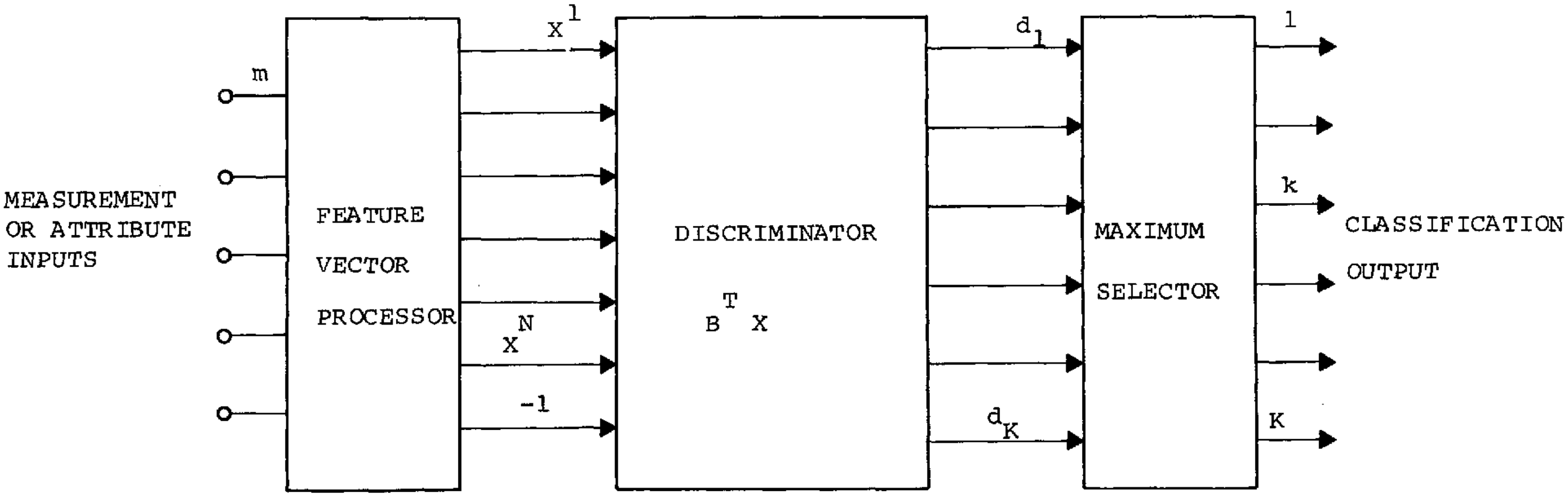


Figure 5. Block Diagram of a Multiclass Signal Classifier (N features, K classes)

$$e = \sum_{j=1}^{K} w_j e_j \qquad (4)$$

where w_j is a weight to be applied to mean square mapping error e_j. The choosing of these weights is done with the Parameter Iteration Method.

The coefficient matrix B is found by minimizing e with respect to the elements of B. If the decision space A is the identity matrix (for simplicity), then

$$B_j = \sum_{i=1}^{K} w_i S_i^{\quad -1} w_j \bar{x}_j ;\ j=1,\ \ldots,\ K \qquad (5)$$

where

$$S_j = E[X_j X_j T]$$

is the feature dispersion matrix for class j and $\bar{X}_j = E[X_j]$ is the mean feature vector for class j.

Because the mapping errors and the distributions of the various classes are different, it would be a coincidence that the linear boundaries between classes determined by the K-Class algorithm would be optimum when all the classes are weighted alike. Thus, we find it necessary to adjust the weights of each class (see Equation 4) to minimize the total mapping error. This is performed by the Parameter Iteration Method.

This method does not actually minimize the mapping error, but does minimize the number of mistakes in the training set of samples, which is surely directly proportional to the mapping error. In addition, a cost parameter is included which will "guard" one class over another. The adjustment of the class weights is

$$w_j^{(t)} = w_j^{(t-1)} + \epsilon^{(t-1)} \frac{\frac{1}{N_j} \sum_{i=1}^{K} c_{ij} m_{ij}^{(t-1)}}{T^{(t-1)}/M} \qquad \sum_{i=1}^{K} w_j^{(t)} \tag{6}$$

where $m_{ij}^{(t)}$ is the number of samples in class j recognized as class i on the t^{th} iteration; c_{ij} is the cost of recognizing class j as class i; N_j is the number of samples in class j;

$$T^{(t)} = \sum_{j=1}^{K} \sum_{i=1}^{K} c_{ij} m_{ij}^{(t)}$$

is the total loss on iteration t; M is the total number of samples, and $\epsilon^{(t)}$ is the variable step size.

Initially, all weights w_j are set equal and the coefficient matrix of Equation 5 is computed and used to test the training set of samples. Based on the testing results, Equation 6 is used to adjust the class weights and the new coefficient matrix is tested. In the CDC 6600 software this is continued while the step size $\epsilon^{(t)}$ is varied until the step size is 0. Anytime the testing results are worse than a previous best, the step size is reduced by a factor δ.

In the DDP 24 software, equation 6 is not divided by $\sum_{i=1}^{K} w_j^{(t)}$: the w_j continue to grow in size with each iteration, while ϵ stays constant. This had the effect of decreasing the step size gradually as the process continues. The real-time capability of the DDP-24 allows this procedure, and allows termination of the iterations at anytime.

In either case the testing is not done on all the samples in order to speed up the process. Only the samples within a band near the classifier boundaries obtained on the first iteration are used for testing. An example of this is shown in Figure 6 for three classes in two-dimensional space. The samples that lie in the shaded area would be saved for testing. The band must be large enough so that the classifier boundaries would not move outside the band during the iteration process. In the CDC 6600 software, the band width is constant; in the DDP-24 software, it is proportional to the statistical distance between classes which is discussed below.

A byproduct of the K-Class algorithm is a distance formula which measures the statistical distance between classes. This formula is much like the well-known Divergence measure. The only difference between the two is that the Divergence measure is based on the likelihood ratio algorithm, while the K-Class distance is based on the K-Class algorithm. The two measures can be derived using the same logical steps. The formula for the distance between class i and j is:

$$D_{ij} = (B_i - B_j)^T \ (\overline{X}_i - \overline{X}_j) = \sum_{k=1}^{N} (b_{ik} - b_{jk}) (\overline{X}_{ik} - \overline{X}_{jk}) =$$

$$\sum_{k=1}^{N} d_{ijk} \qquad (7)$$

where D_{ij} is the distance between classes i and j; d_{ijk} is the component attributed to feature k.

The d_{ijk} are not independent of each other; however, they can be a good measure of how much feature k adds to the total distance. There are many different procedures that can be used to select features with this measure. The programs discussed here only print out the D_{ij} and the d_{ijk}.

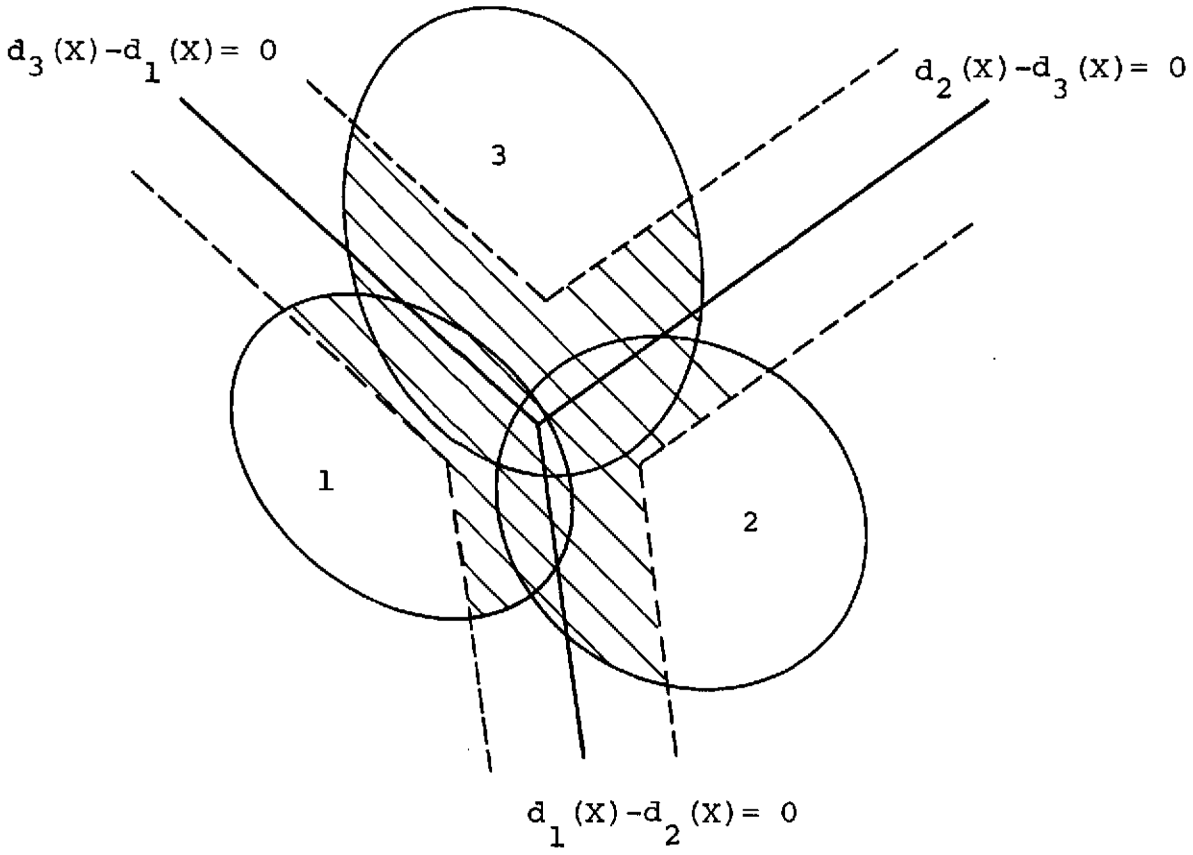


Figure 6. Illustration of Samples Saved for Use in the Parameter Iteration Method

The above theory can be used in a systematic approach to determine a best set of features and the optimum of K-Class boundaries for that set of features. Figure 7 presents the flow of events of this systematic approach.

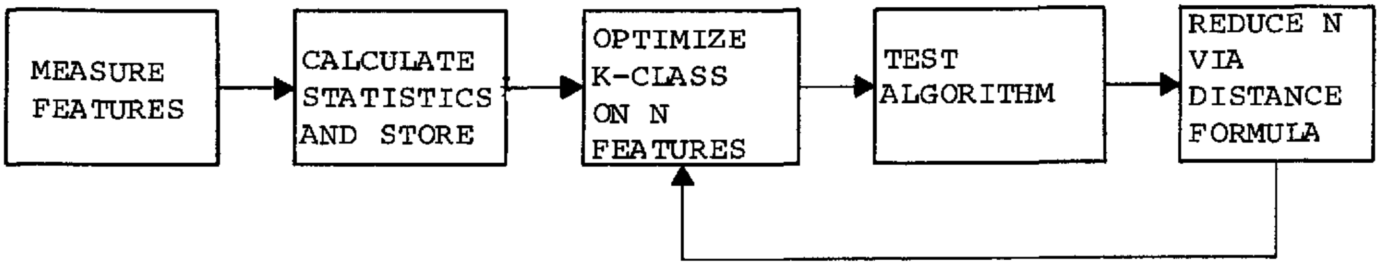

MEASURE FEATURES
CALCULATE STATISTICS AND STORE
OPTIMIZE K-CLASS ON N FEATURES
TEST ALGORITHM
REDUCE N VIA DISTANCE FORMULA

Figure 7. Flow Chart for Optimizing K-Class

In addition to the distance measures between classes indicating the effectiveness of the various features for distinguishing between classes, K-class also prints out a confusion matrix. This is an array showing the correct assignments as rows and the K-class assignments as columns. From this confusion matrix, one can determine the overall performance as well as the classes causing the greatest problems.

The confusion matrix can also be expressed in percentage terms. If the class membership is approximately equal, percentages are easy to interpret, however if the sample sizes are vastly different, the percentage confusion matrix can be misleading. For example, one can tolerate large percentage misclassifications if the class size is small because only a few actual misclassifications will be involved.

The confusion matrix produced by K-Class are the results obtained when training and testing on the same data set. Techniques are available for testing performance on data that was not used for training. The procedure we used on the Cloquet Forest Test Site was to cross correlate the ground truth map with the thematic map generated by K-Class. A second procedure, which is easier to implement, selects points at random from the various classes and determines the K-Class assignment for these points. This approach alleviates the precise registration requirement for cross correlating the ground truth and K-Class output map since the test points can be selected from within the class boundaries away from the borders. The disadvantage of this approach is that it avoids boundaries where class mixtures are most likely to occur, thus the confusion matrix is optimistic.

Thematic Map Generation:

After the weights have been determined in the training procedure, the production of a thematic map is relatively easy. A block diagram of a three-class, four-feature discriminator is shown in Figure 8. The class structure and features must be the same for training and producing the thematic map. The discrimination process simply involves a multiplication of the augmented feature vector by a set of weights for each class. The class assignment is made by a maximum selector placing the input sample in the class having the maximum decision number. The sum of the three decision numbers is unity so that at least one of them is positive. The relative magnitude of the decision number serves to indicate the similarity between the sample from which the features were generated and the training sets for the various classes.

Clustering Algorithm

The difference between supervised and unsupervised learning is the presence or absence of an assigned class structure for data samples. K-Class uses a training set which has been assigned to classes by a photointerpreter.

In addition to K-Class, we used the ISODATA clustering routine for the Cloquet test site data. This is unsupervised learning where parameters of the cluster groups are estimated in an iterative procedure. The basic procedure for ISODATA is shown in the block diagram in Figure 9. The procedure

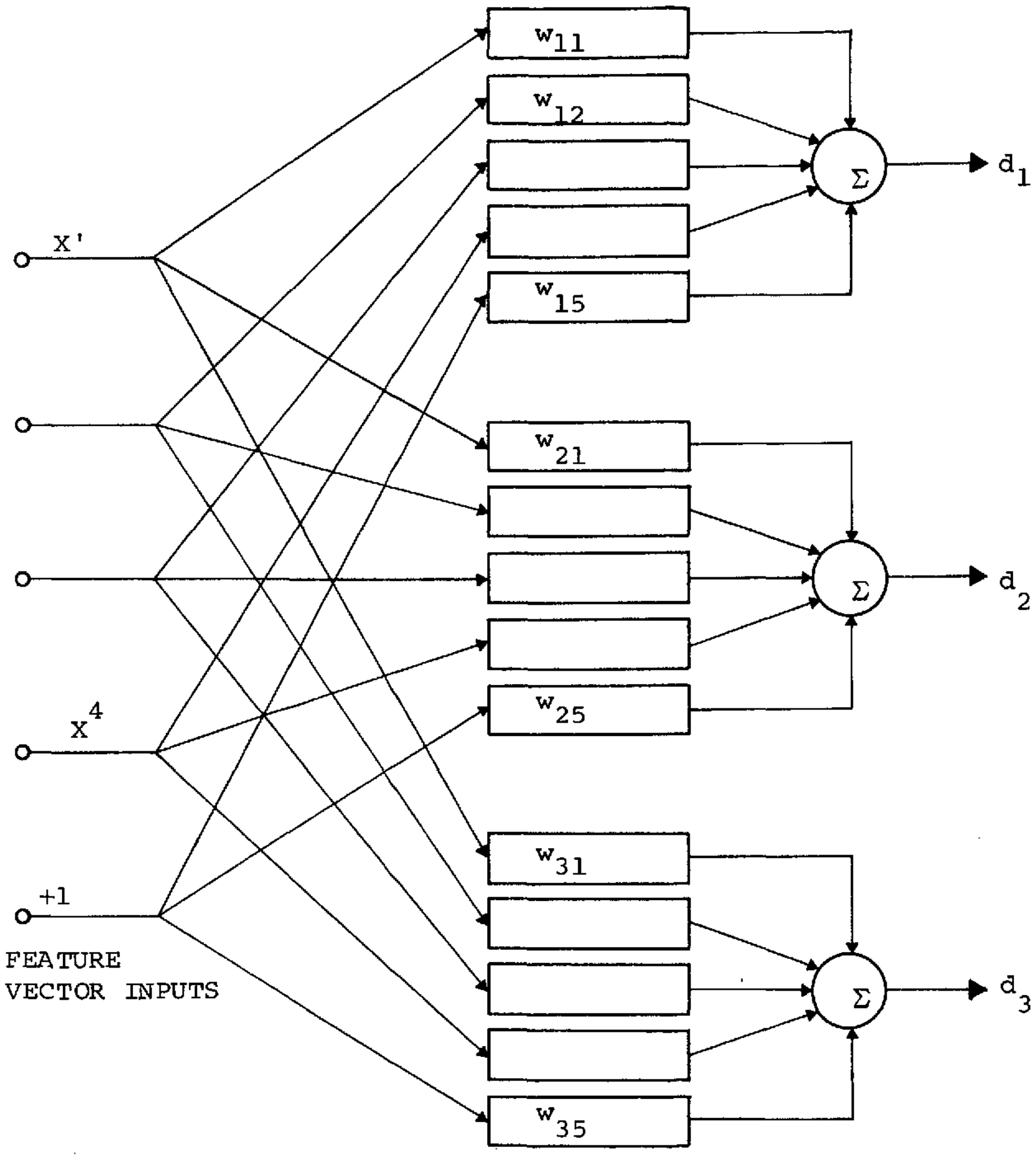


Figure 8. Block Diagram of a Three-Class, Four-Feature Discriminator

sketched in Figure 9 does not include the more detailed provisions for breaking ties, throwing out wild shots and more sophisticated terminating conditions - it does however display the essential concept. The result of the algorithm is a fit to the data of a set of cluster centers that tends to minimize the sum of the squared distances of each data print from its closest cluster center.

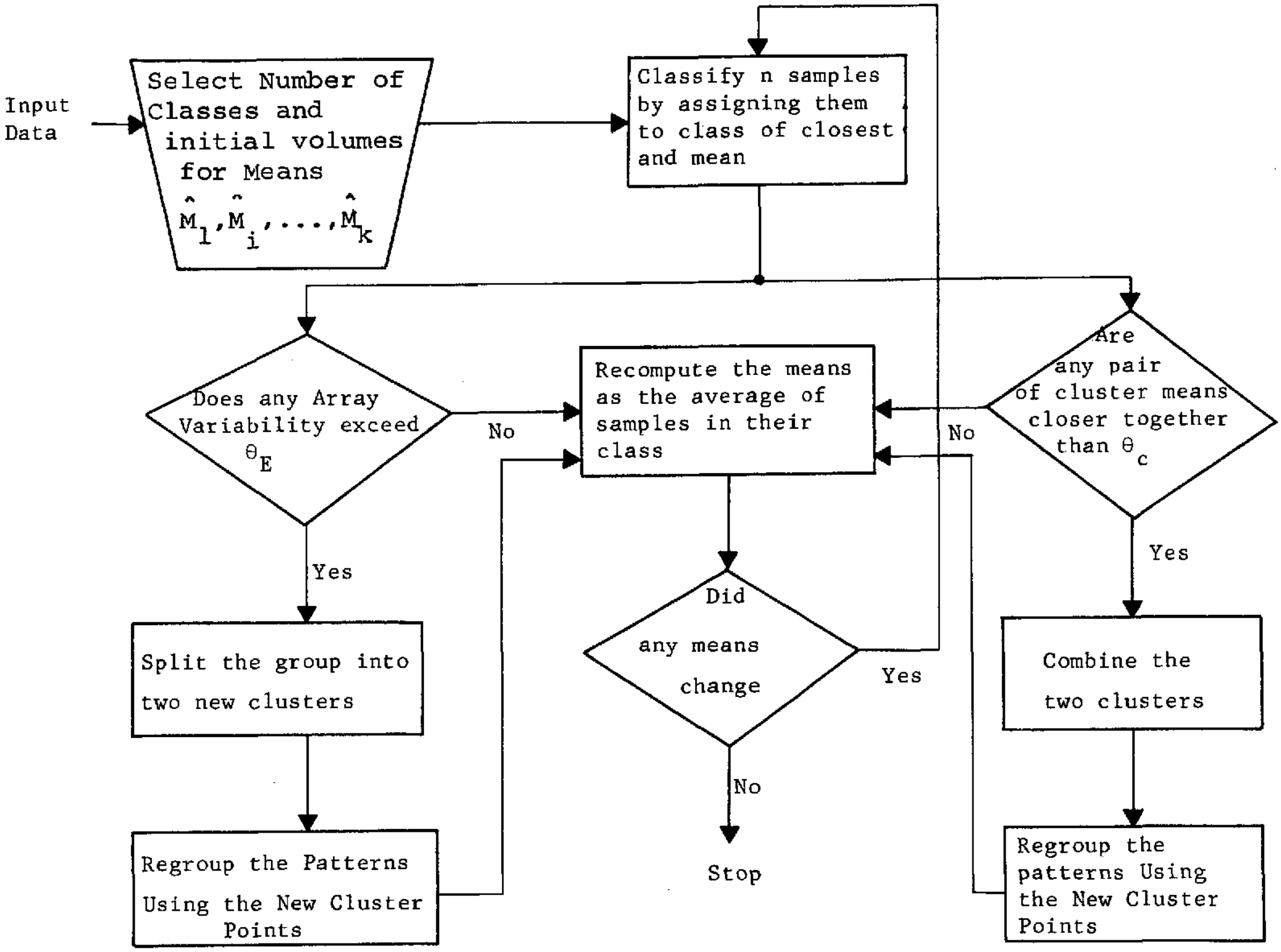
The output from such a procedure can be used as a thematic map. The clusters must be labeled as specific classes to be of any value. This labeling can be on the basis of similarity between cluster means and the means of known classes. An alternative approach would be to compare the ISODATA output thematic map with a ground truth map generated by a photointerpreter. This comparison is an invaluable aid in checking the accuracy of the ground truth map.

Results:

All of the features used for automatic classification of the Cloquet Test Site were derived from the four MSS bands of ERTS-A image 1075-16312. This is an October 6, 1972 coverage that is relatively cloud free, however a slight amount of haze is visible over the test site as seen from the NASA Color Combined photo. The data was extracted from the 7-track 800 BPI Computer Compatible tapes.

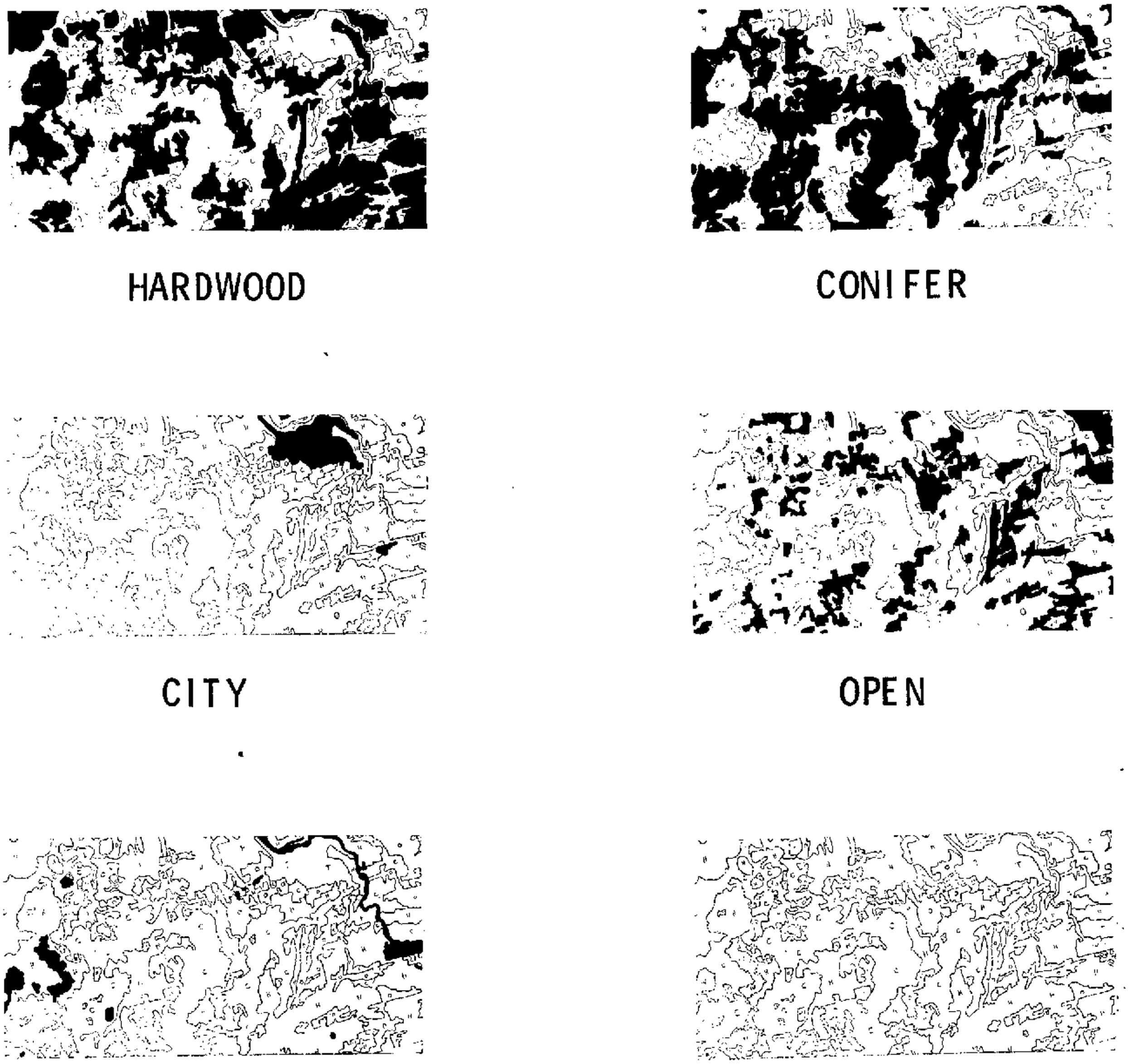
Multispectral and spatial features were used for automatic classification. The most effective bands for separating the five classes listed above were determined by using the principal components algorithm. MSS band 6 had the greatest effectiveness followed by MSS 7.

The Cloquet area was delineated into five classes: conifers, hardwoods, open, water and urban. An estimate of the relative amounts of the five classes as shown in Figure 10 is listed in the table below:


Input Data
Select Number of Classes and initial volumes for Means $\hat{M}_1, \hat{M}_i, \ldots, \hat{M}_k$
Classify n samples by assigning them to class of closest and mean
Does any Array Variability exceed θ_E
No
Yes
Split the group into two new clusters
Regroup the Patterns Using the New Cluster Points
Recompute the means as the average of samples in their class
Did any means change
Yes
No
Stop
Are any pair of cluster means closer together than θ_c
No
Yes
Combine the two clusters
Regroup the patterns Using the New Cluster Points

ISODATA

Figure 9. Clustering Algorithm



HARDWOOD

CONIFER

CITY

OPEN

WATER

Figure 10. CLOQUET GROUND TRUTH MAP

Table I Cloquet Site Class Structure

Conifer	32%
Hardwoods	38%
Open	20%
Water	5%
Urban	5%

Approximately a thousand acres of each class were used for training.

Using only multispectral features, the classifier was trained on data derived from the 8 x 8 sample arrays from each class. Each training sample consisted of one data point or pixel of approximately one acre. When the classifier is trained on a data set, a set of weights are obtained which will cluster samples of each class around the points of a simplex in a least mean square sense. If one applies these weights to the training data, its performance is shown by the confusion matrix in Table II.

Seventy four percent of the data points are correctly classified. An indication of the types of mistakes made are shown in the confusion matrix. Notice that confusion is most common where "city" is called "open", the next most common is "open" classed as "hardwood" and then "open" classed as "city". These confusions appear reasonable since the classification was performed using October data when the hardwoods have shed their leaves and grasslands are becoming dormant. Therefore, one would expect open, hardwood and city to look alike. This is further substantiated by noting the overlaps in the density histograms in Figure 3.

The performance using only multi-spectral data is shown also in Figure 11. When using individual data points, i.e., the 4 MSS density bands as features, performance is 74%.

Table II. Automatic Classification Results Based on Density - Cloquet Test Site

	Type	Sample Size	Automatic Classification Assignments in Percent				
			Hardwood	Conifer	Open	Water	City
Ground Truth Assignments	Hardwood	896	80.7	4.1	4.9	1.8	8.5
	Conifer	1152	5	80.9	2.3	13.9	2.4
	Open	960	20.4	3.5	61.6	.6	13.9
	Water	640	10.2	.0	.3	88.6	.9
	City	640	8.1	6.1	24.1	6.4	55.3
	TOTAL	4288	Correct Classification Assignments = 3167				

An indication of the effect of adding texture to multispectral features is shown by the curve in Figure 11. Texture was computed from Band 7 using the Slant Transform. The Slant Transform as previously described, is an image transform with a basis vector matched to the gradual brightness changes along an image line which compacts the image energy to as few of the transform domain samples as possible. When computed on a 4 x 4 array and added to the density features, the performance increases to 90%. If computed on an 8 x 8 array and added to density, the performance increases to 99%. These results were obtained by testing on the training set.

In comparing the texture algorithms, it was found that the Karhunen Loeve transform provides the highest classification accuracy. For 8 x 8 arrays, the Slant Transform outperforms both the Walsh and Fast Fourier as shown in Figure 12. As the dimensionality is increased, the Fourier transform performance is better than either the Walsh or Slant transforms. The Fourier transform asymptomatically approaches the Karhunen Loeve transform at large dimensions.

After obtaining the training weights on the Cloquet test site, they were used to generate a delineation map for 24,000 acres including the training areas. The results of the automatic classification are shown in Figure 13. This output

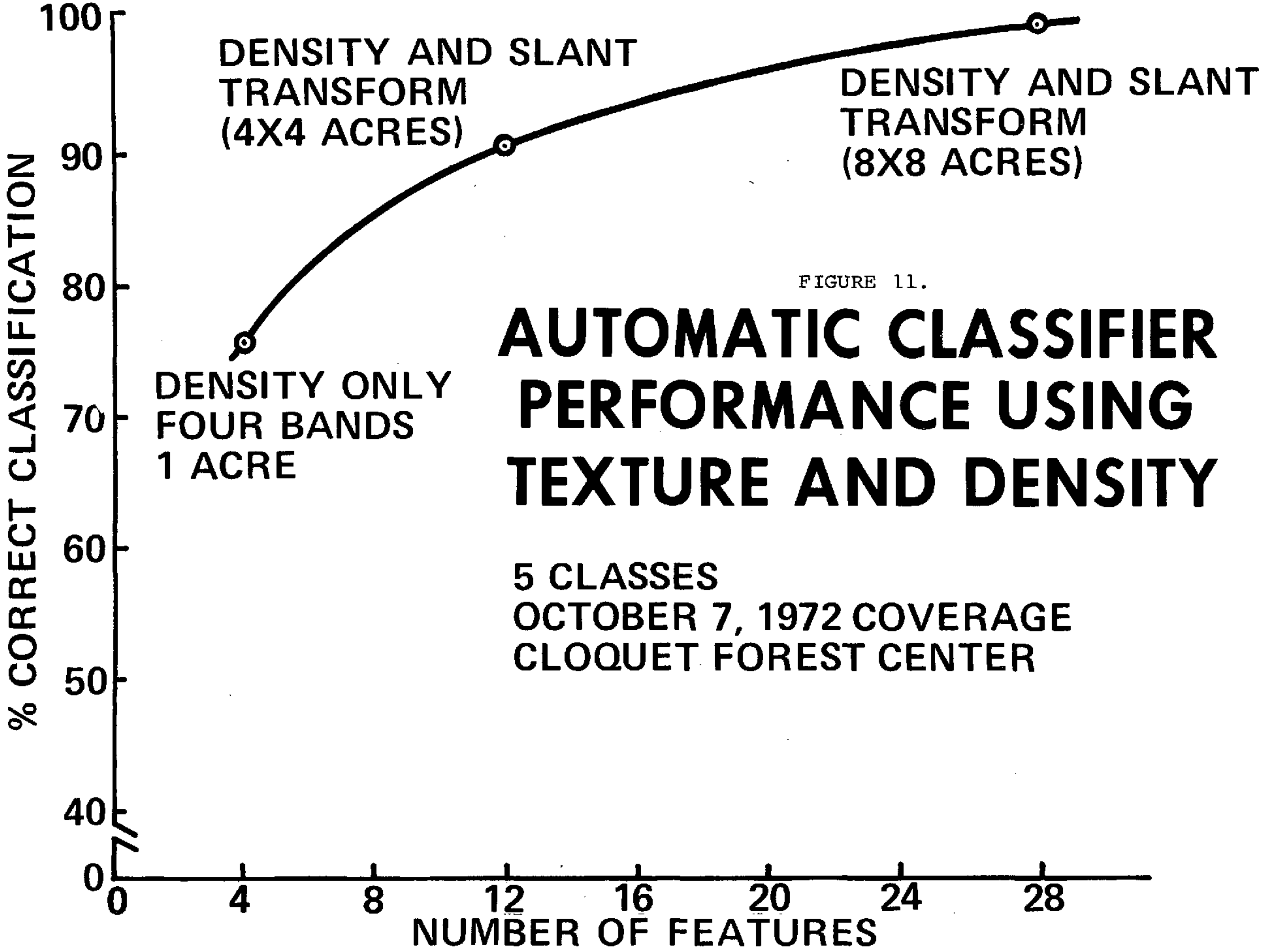


FIGURE 11.

AUTOMATIC CLASSIFIER PERFORMANCE USING TEXTURE AND DENSITY

FIGURE 12.

COMPARISON OF TEXTURE ALGORITHMS FOR AUTOMATIC CLASSIFICATION

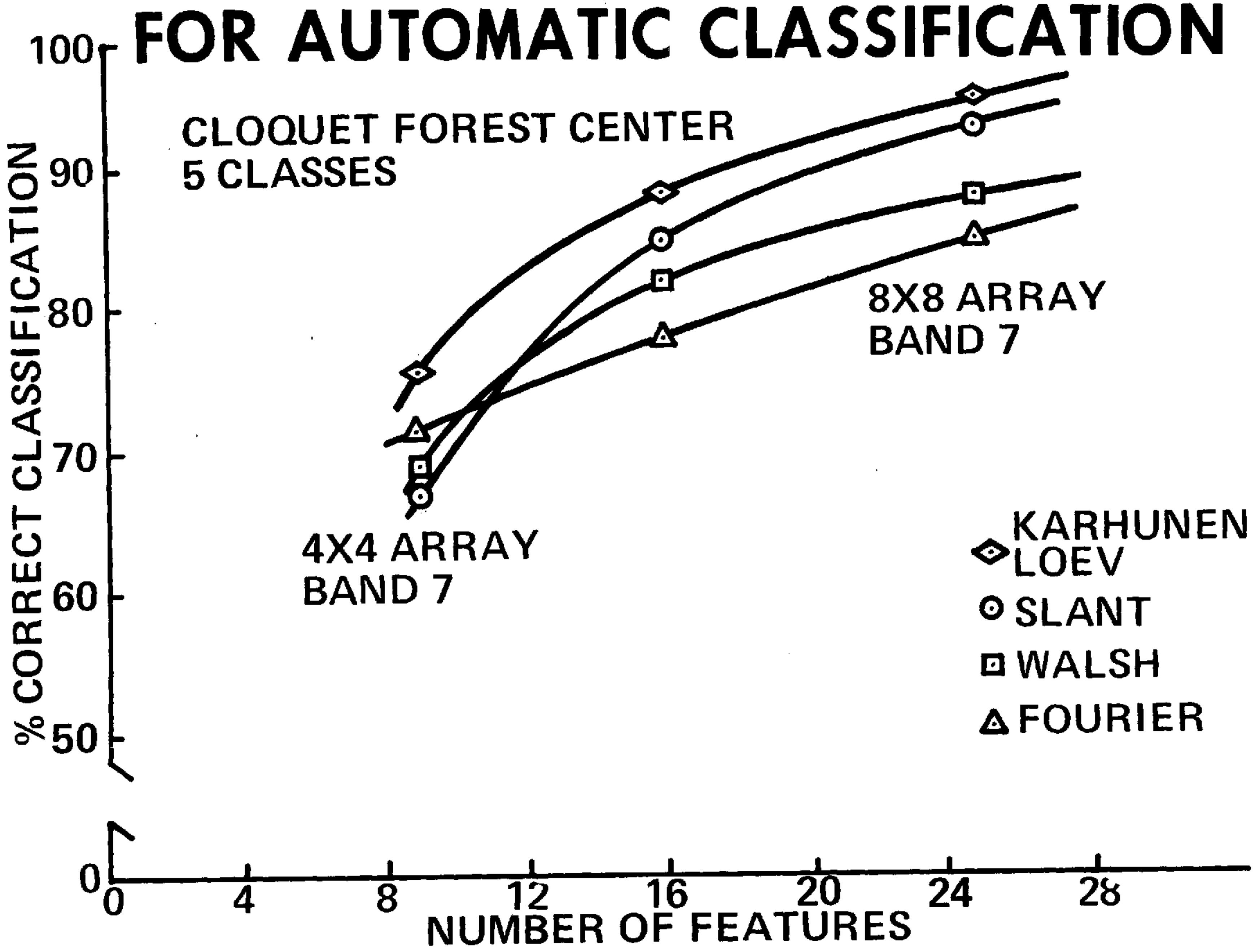


FIGURE 13. K-CLASS OUTPUT THEMATIC MAP OF CLOQUET FOREST CENTER AREA, p. 28

map can be visually compared with the ground truth map in Figure 11. In the generation of the thematic map, each input data print is assigned to a class on the basis of its distance from the orthagonal vector points, these points are assigned to the various classes during training. The density levels in the photo are assigned to the five classes in the following manner: hardwood, conifer, urban, open, water (going from light to dark).

The alternative technique which has been used to group the unknown data is based on natural clusters. The clustering technique is very useful for checking ground truth used for training a classifier. Errors in the training set are very serious. These become obvious because they do not fall into natural clusters. Clusters can also be used to make delineations which can then be assigned to the various ground truth classes. This is useful when class designations are still being invented. The Cloquet data from MSS bands 6 and 7 was clustered into eight clusters. These were assigned to one of four classes based on the distance of the cluster centers from the means of samples of data from the four classes. A thematic map of Cloquet is shown in Figure 14 where water is blue, hardwoods-yellow, conifers-green and open-yellow.

Chippewa National Forest Test Site:

Our next objectives were to extend the procedures evolved from the small Cloquet test site to a much larger area providing greater statistical significance. In addition, we wanted to work with an area that could be isolated as a management unit to promote user interaction. We also wanted to find an area with enough samples of a variety of species. And of course, the final criterion was the availability of RB-57 over-flights of the area for ground truth and cloud free ERTS coverages, preferably over a number of seasons.

Considering only the first two constraints, either Superior National Forest or Chippewa National Forest would suffice. However, Chippewa has a good mix of hardwoods and conifers whereas Superior is primarily conifers. Either forest contains over one million acres.

The first cloud free image of the Chippewa National Forest was obtained on October 7, 1972 (1076-16370). A winter coverage of the area was recorded on January 5, 1973 (1166-16373). These two coverages were used as the source of data for our classification.

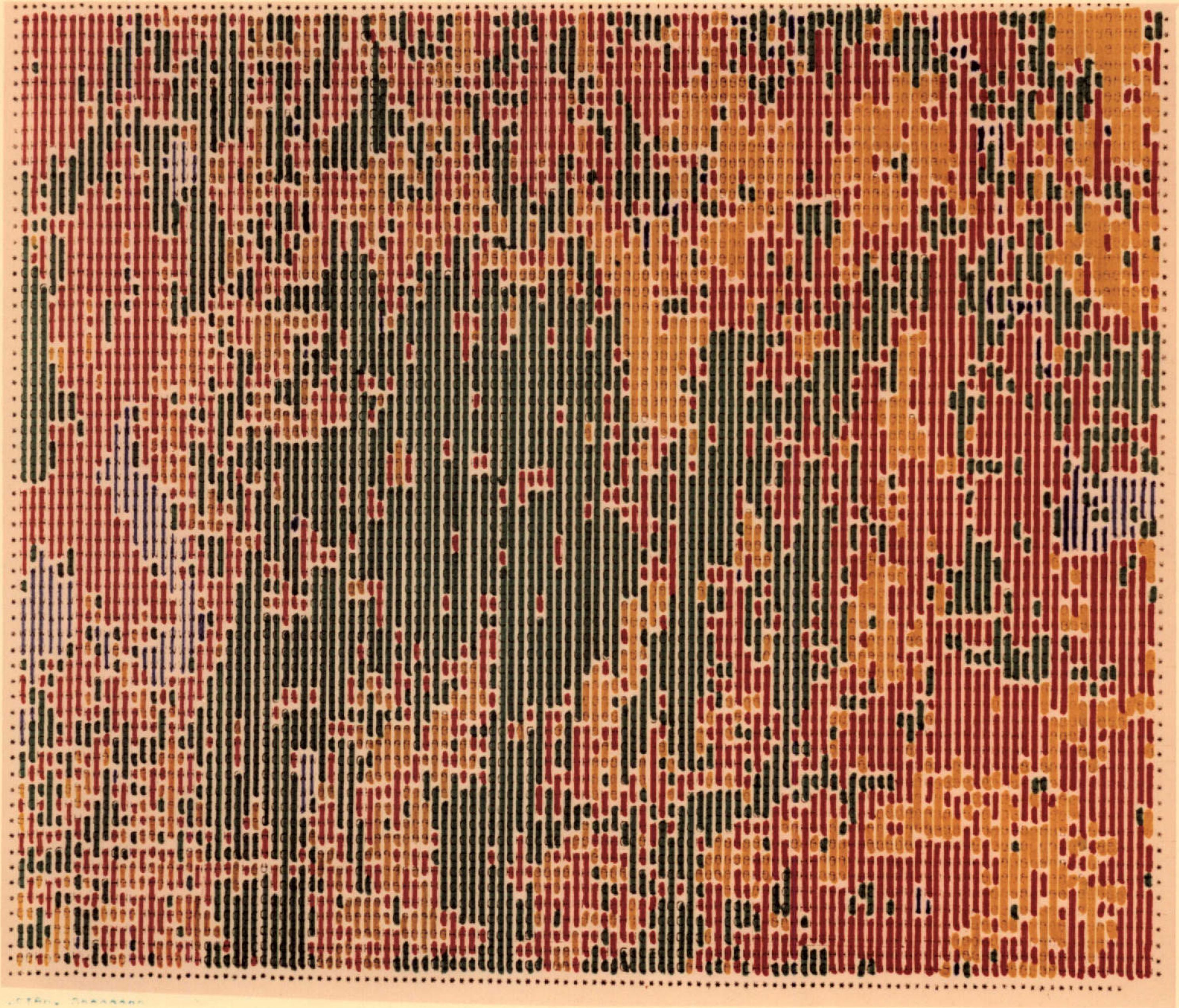


FIGURE 14. CLUSTERING ALGORITHM THEMATIC MAP, p. 30

Class Selection

The classes selected for the Chippewa National Forest are as follows: open, water, marsh, cutover, and twelve forest types, namely: hardwood, conifer and mixed (containing more than 25% of both varieties); these three types are further delineated into upland or lowland and finally into high and low crown density (above or below 50%). The selection of these classes was based on the typical class structure used by foresters in the area. The species included in the broad cover types are listed below:

Table III Chippewa National Forest
Cover Types

	Upland	Transition	Lowland
Conifer	Jack pine	Balsam fir	Black spruce
	Red pine		Tamarack
	White spruce		Northern white cedar
	White pine		
Hardwood	Trembling aspen	Green ash	Black ash
	Paper birch	American elm	Balsam poplar
	American basswood	Yellow birch	Silver maple
	Sugar maple		
	Big tooth aspen		
	Red oak		

Ground truth for the Chippewa National Forest was obtained from the June 6, 1972 RB-57 overflight. The Color IR RB-57 aerial photos at 1:60,000 were used to delineate approximately 220,000 acres, out of which about 25,000 acres were used for training on the various classes as shown in Tables IV and V.

The ground truth map is shown in Figure 15. Training samples were selected from each of 14 classes as indicated in the sample size summary table. Again, the training samples were

Table IV Chippewa National Forest

TRAINING
SAMPLE SIZE SUMMARY

CLASS*	West Half-CCT2		East Half-CCT3		Total		Total # Pixels
	8x8	4x4	8x8	4x4	8x8	4x4	
2		10		3		13	208
3	37		1		38		2432
4	13		4		17		1088
5		13		1		14	224
Ah"**	75		1		76		4864
As"	35		1		36		2304
As'	20				20		1280
Bh"	64		27		91		5824
Bh'	3		3	19	6	19	688
Bs"	7		4		11		704
Bs'		6		2		8	128
Mh"	41		3		44		2816
Ms"	28				28		1792
Ms'	7	10		4	7	14	672
TOTAL							25024

* See Table V for Coding Key

**There were no areas with Ah' or Mh' and very few areas of mixed residential.

Table IV Chippewa National Forest - Cont.

TOTALS

Class	Pixels	% of Training Set
Non forest	3952	15.8
Conifer (A)	8448	33.8
Hardwoods (B)	7344	29.3
Mixed (M)	5280	21.1

Total acreage delineated = 220,425

Training size 11.35% of total

Table V Chippewa Coding

2	OPEN - PASTURE - CLEARCUTS
3	WATER
4	MARSH
5	CUTOVER
8	MIXED RESIDENTIAL (this class was not included in the training set because of the lack of samples)

FOREST

TYPE	SITE	DENSITY
A-CONIFER	h-upland (highland)	" - >50%
B-HDWD	s-lowland (swamp)	' - <50%

FIGURE 15. CHIPPEWA NATIONAL FOREST

GROUND TRUTH MAP

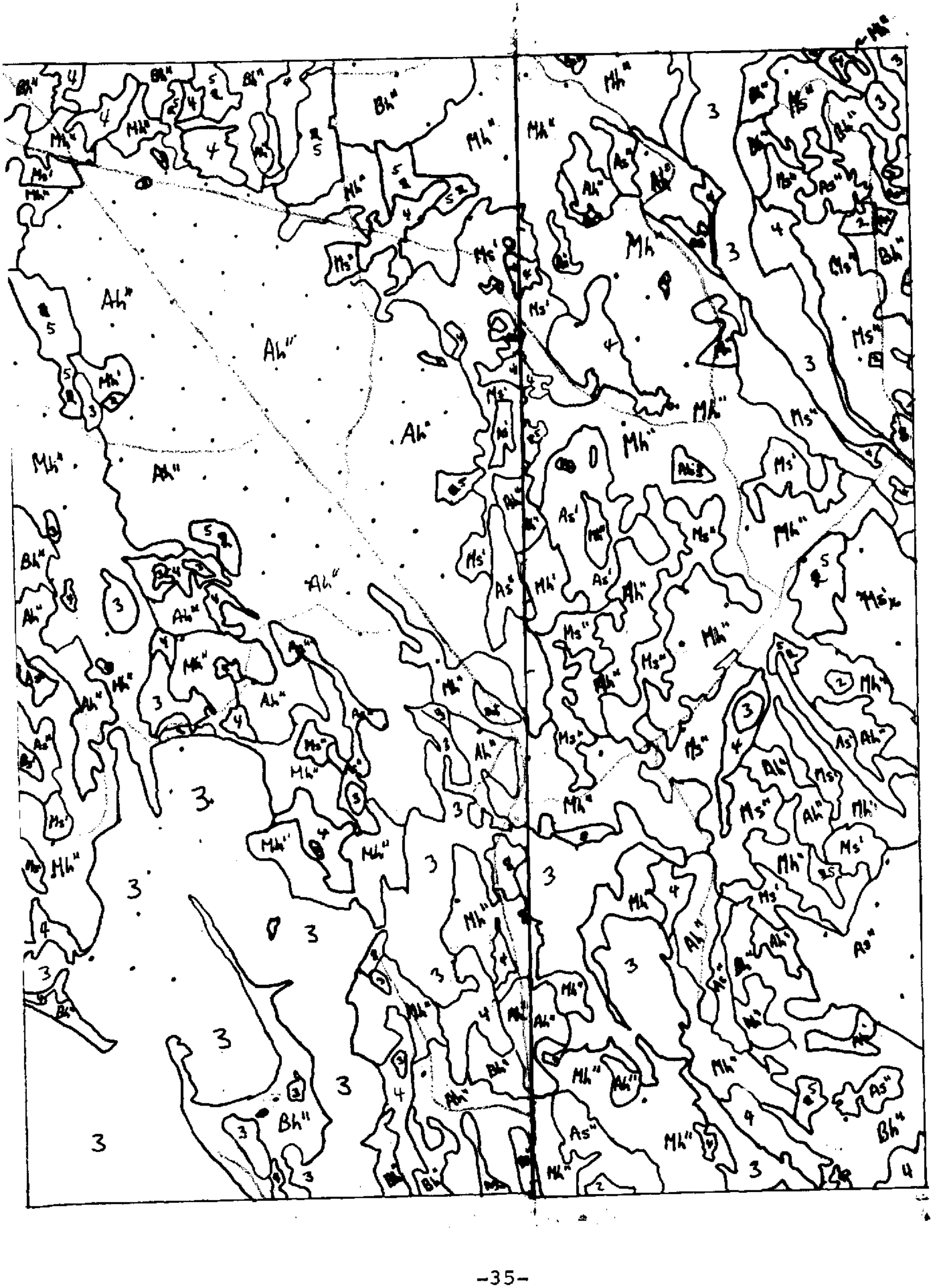


FIGURE 15. CHIPPEWA NATIONAL FOREST

GROUND TRUTH MAP

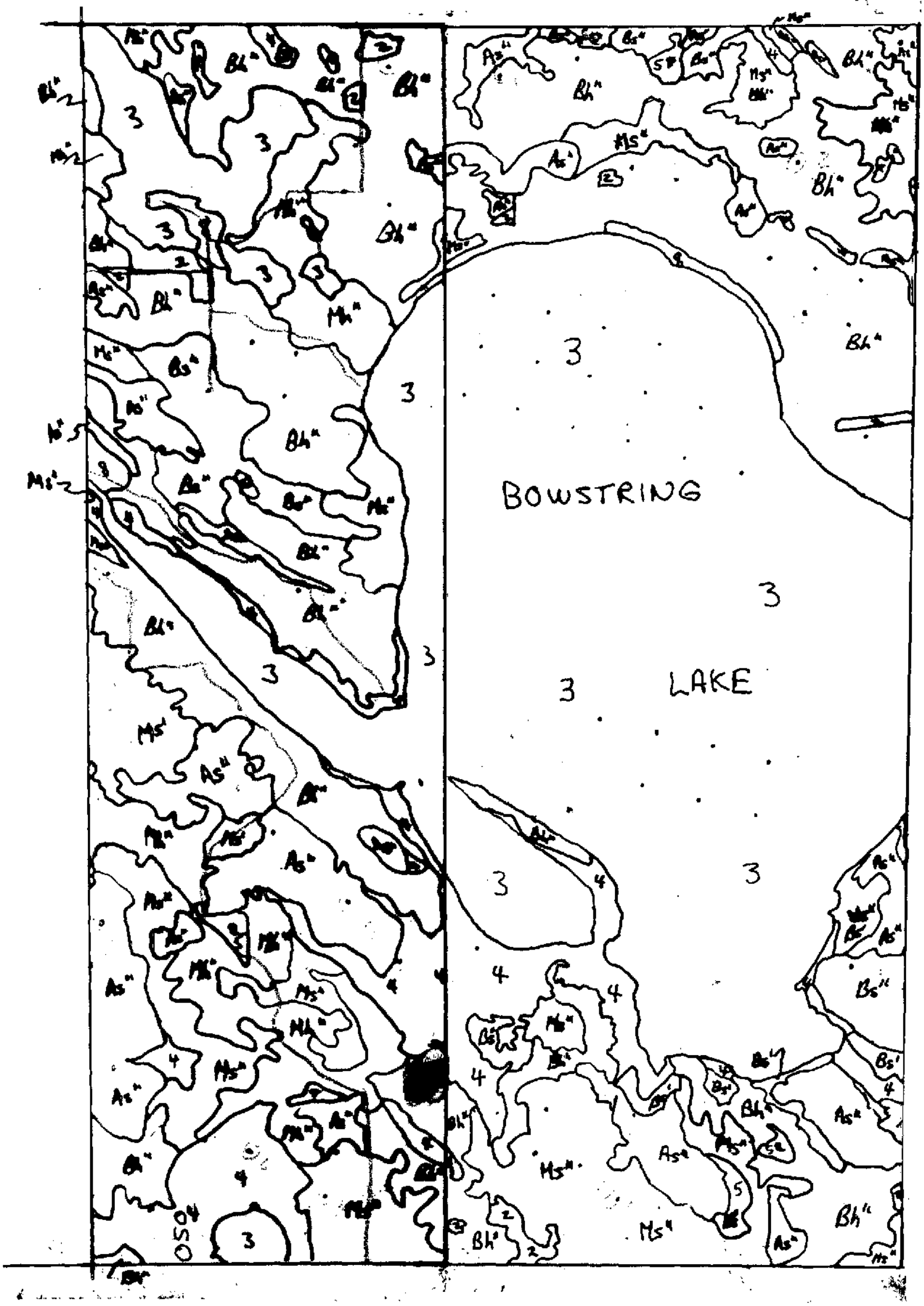


FIGURE 15. CHIPPEWA NATIONAL FOREST

GROUND TRUTH MAP

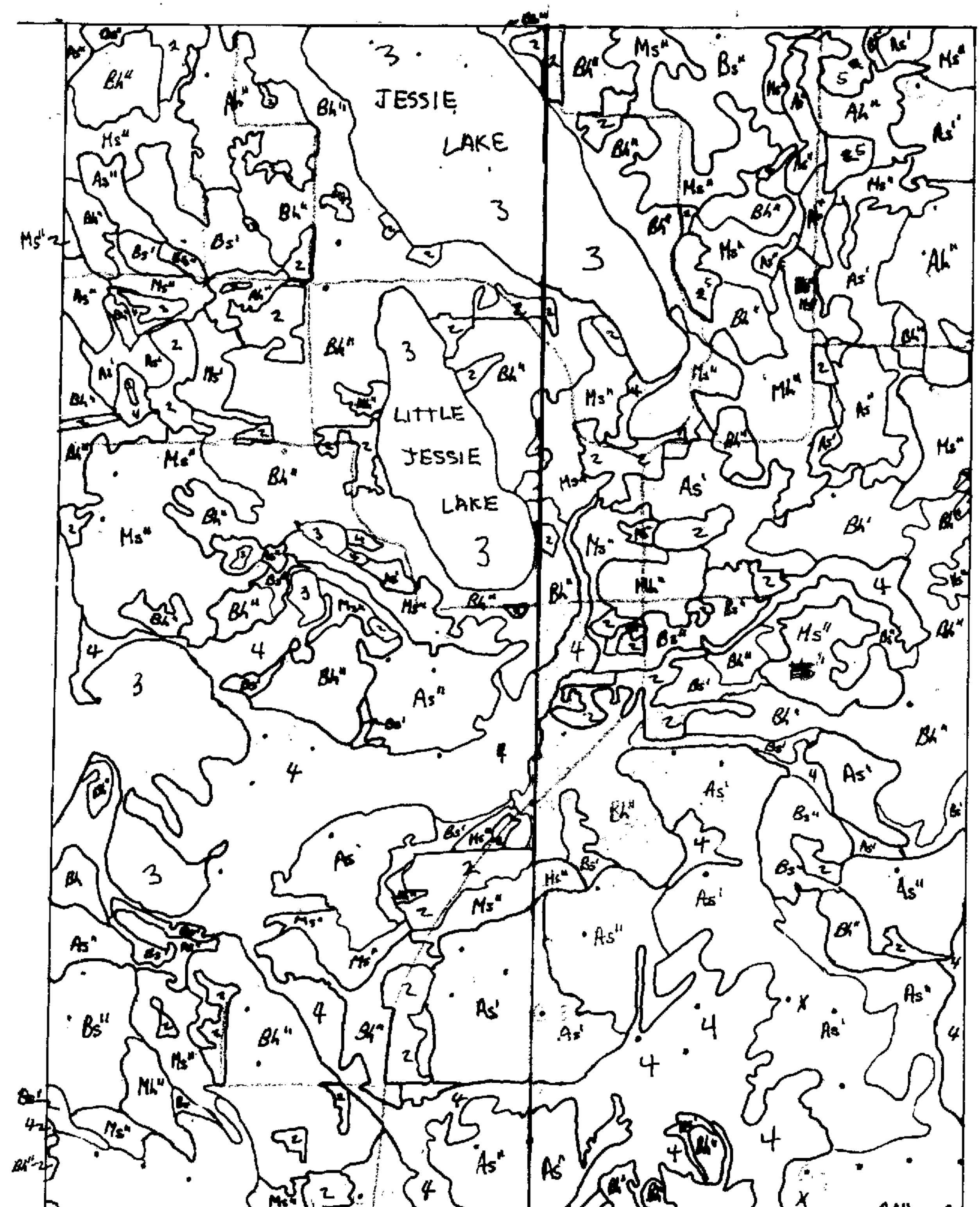


FIGURE 15. CHIPPEWA NATIONAL FOREST

GROUND TRUTH MAP

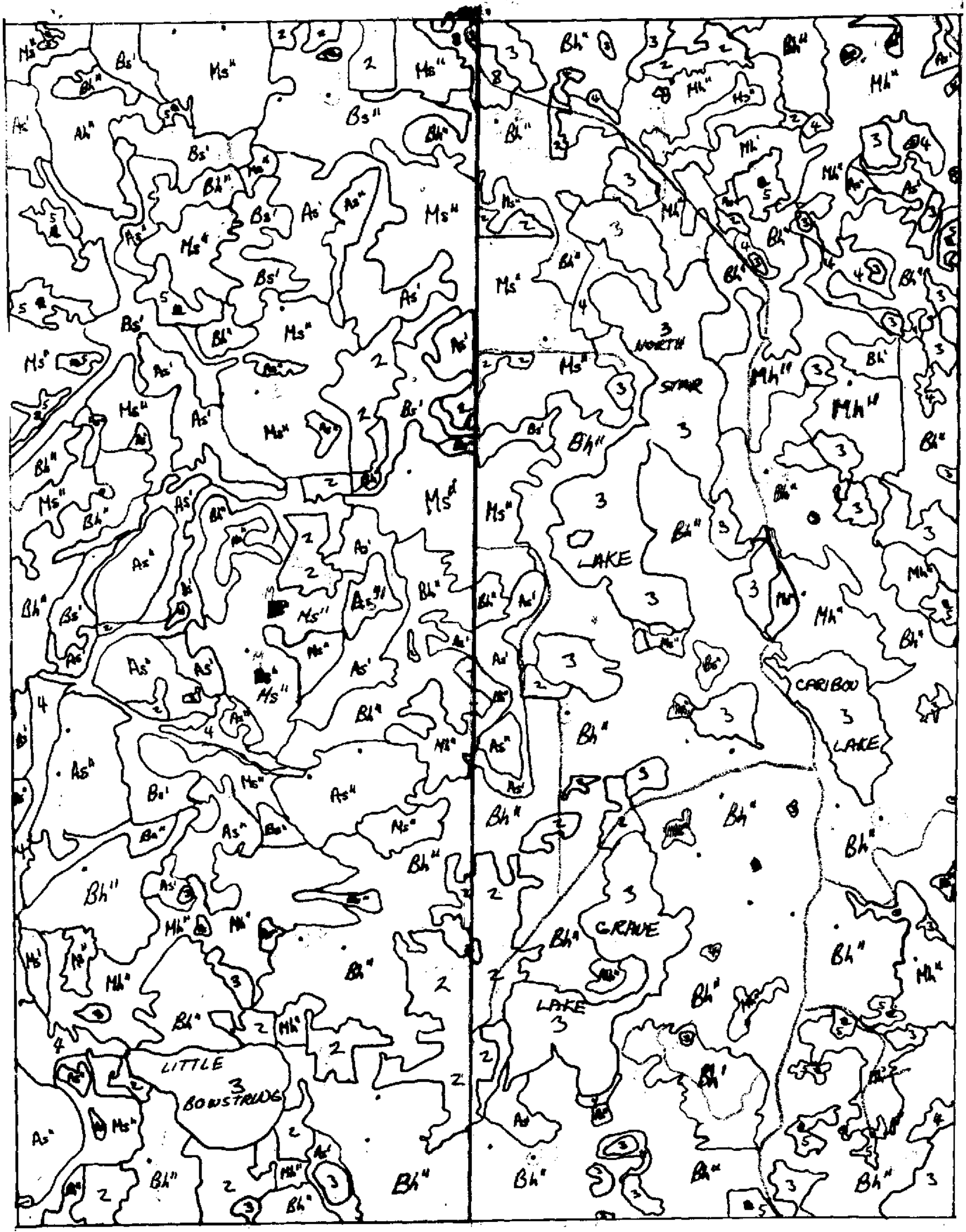


FIGURE 15. CHIPPEWA NATIONAL FOREST

GROUND TRUTH MAP

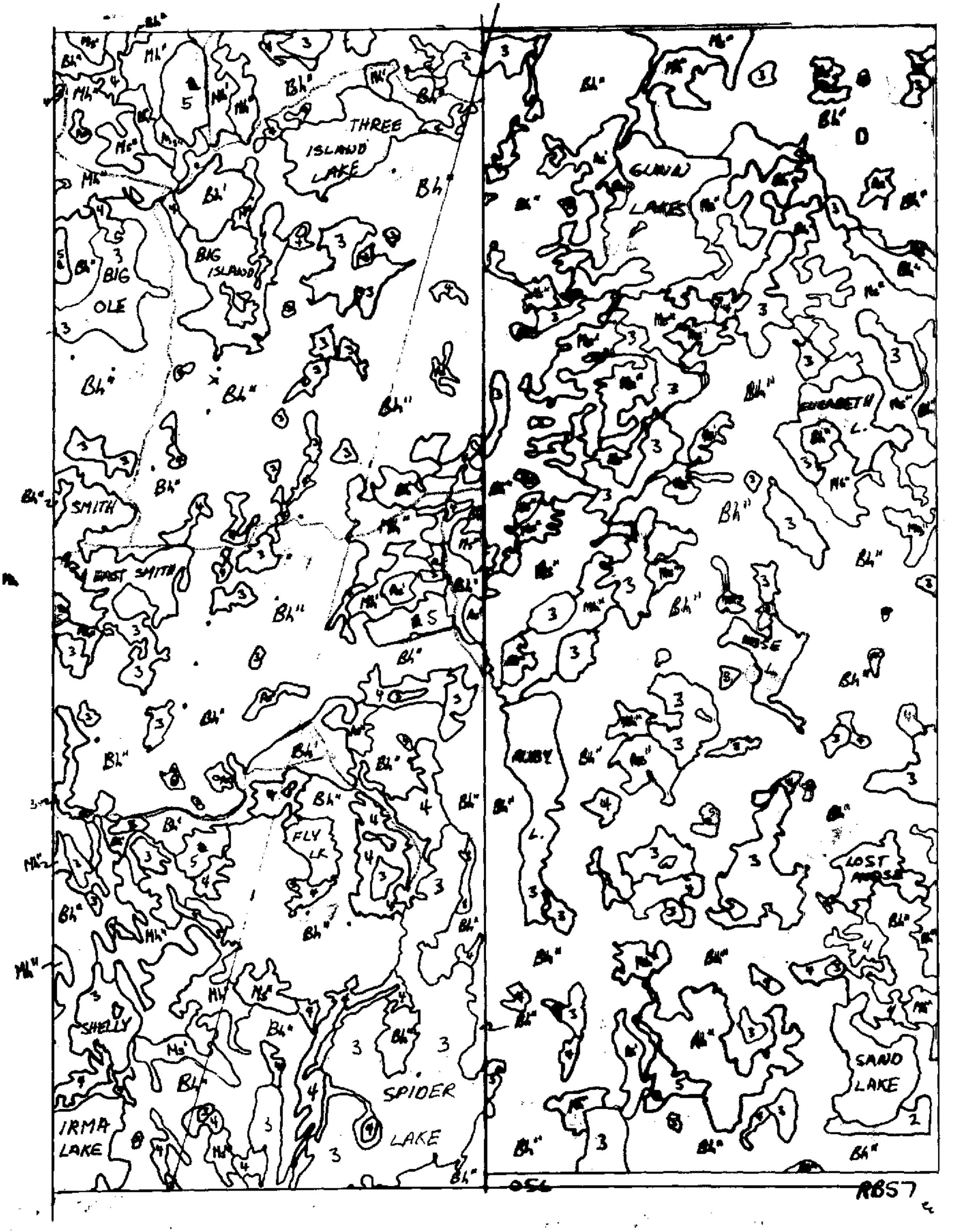


FIGURE 15. CHIPPEWA NATIONAL FOREST

GROUND TRUTH MAP

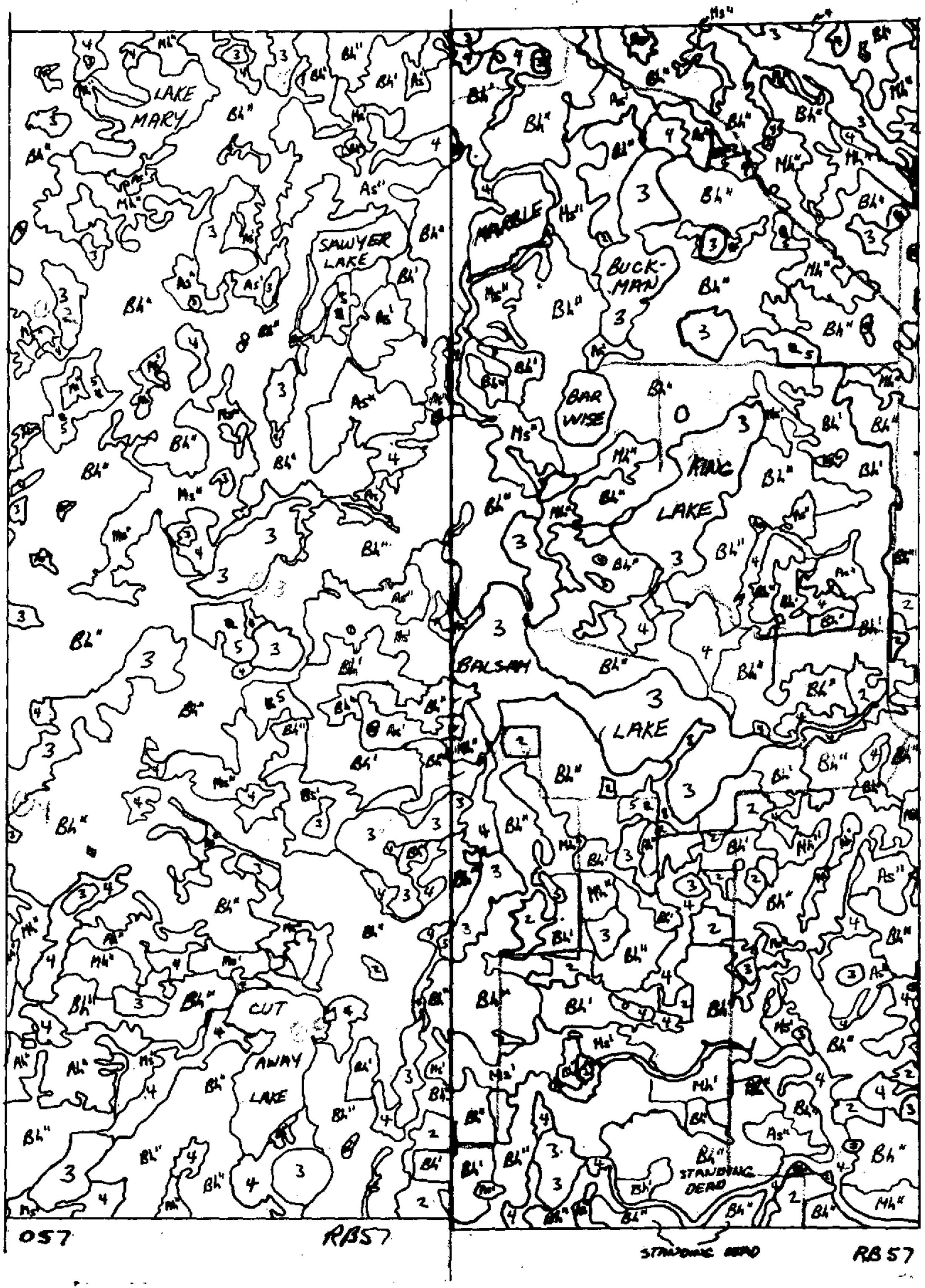
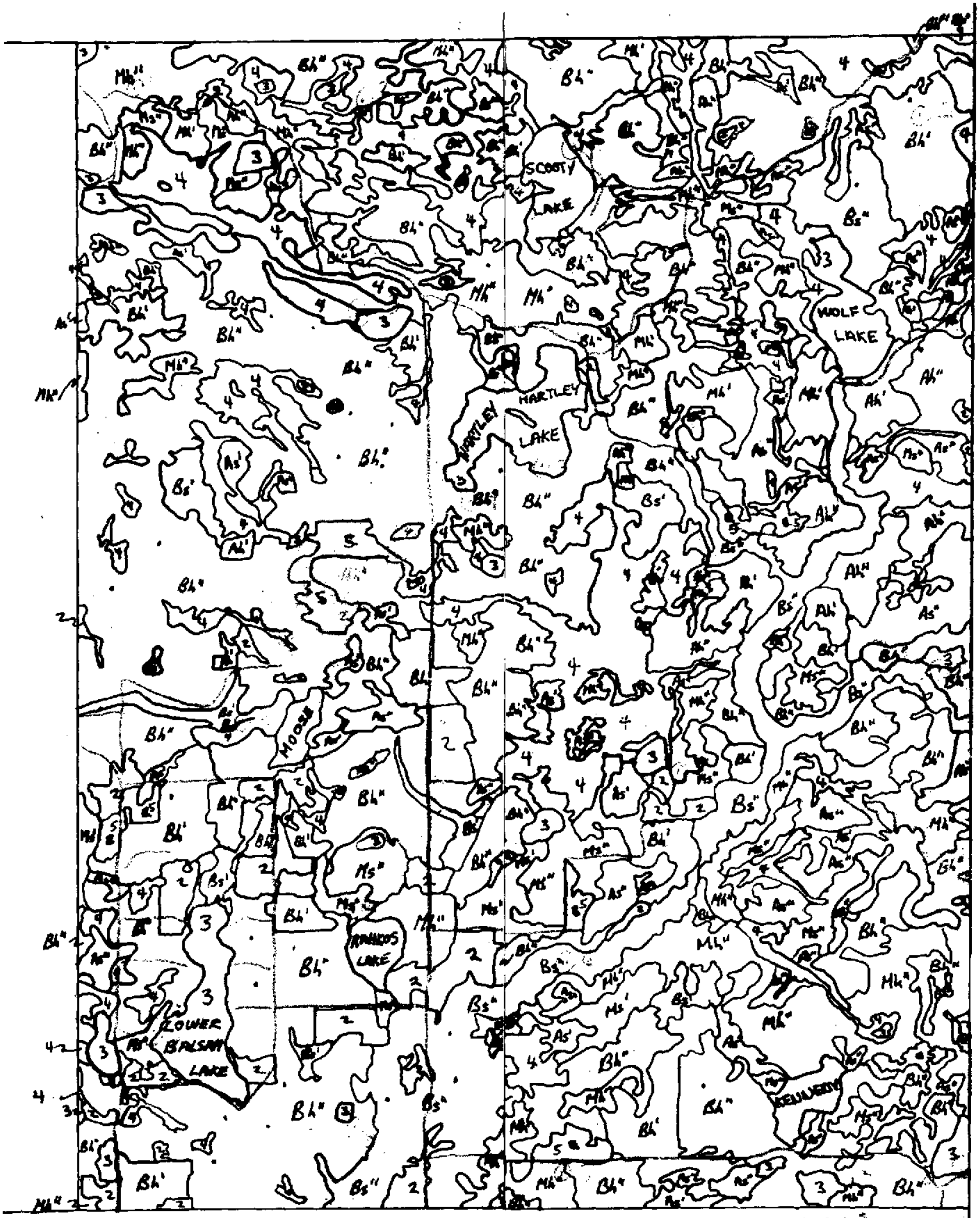


FIGURE 15. CHIPPEWA NATIONAL FOREST

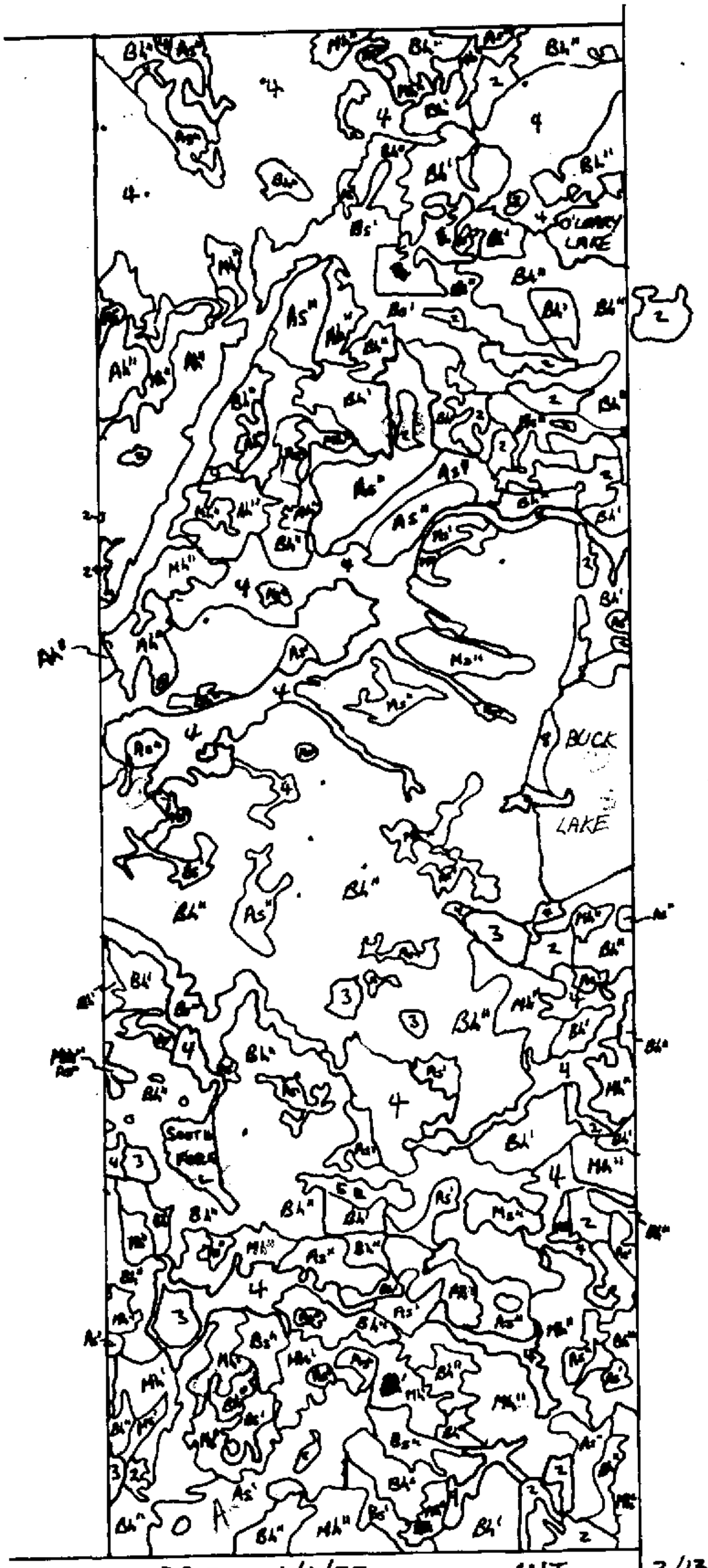
GROUND TRUTH MAP



RB 57 6/6/72
INT. 2/12/73 GRJ

FIGURE 15. CHIPPEWA NATIONAL FOREST

GROUND TRUTH MAP



RBS7 6/6/72 INT 2/13/72 GRJ

comprised of 8 x 8 and 4 x 4 arrays selected from the central positions of the areas containing the various classes. Special attention was given to avoid approaching class boundaries for two reasons, first to avoid a maxture of classes and secondly, to make allowances for sight errors in registration.

Results

Features for automatic classification of the Chippewa Forest were derived from two cloud-free coverages on October 7, 1972 and January 5, 1973. The ERTS-1 frames are 1076-16370 and 116-16373 respectively.

The first classification runs were performed using density features only from the October 7, 1972 coverage. An additional run was made using the October and January data to indicate the improvement in performance achieved by the use of multi-temporal data.

Band 4 on both coverages contained a great deal of banding noise. Because of this extraneous noise,runs were made with and without this band. When using Bands 5, 6 and 7 from the October coverage and five classes, the overall performance was 61.3%. Adding the January coverage from bands 5, 6 and 7 improved the performance to 72.7% as shown in Table VI.

When all eight bands were used from the October and January coverages, the overall performance dropped to 71.3% indicating that band 4 does indeed introduce noise and degrades performance.

The Conifer and hardwood classes were broken down into five classes by sequentially classifying each class into sub-classes as shown in Figure 16. The classification was performed for bands 5, 6 and 7 from October only and for October plus the January coverage. The confusion matrices are given in Table VII.

Table VI Chippewa National Forest Confusion Matrices

6 FEATURES BANDS 5,6,7 FROM OCT. AND JAN. COVERAGES

		Classifier Output					
Ground Truth	Open	92.8	2.9	4.3	0.0	0.0	0.0
	Marsh	4.2	45.8	25.8	11.2	8.7	4.2
	Cutover	9.2	25.0	60.5	0.0	3.9	1.3
	Conifer	0.0	0.1	0.0	87.5	2.7	9.7
	Hardwood	0.4	1.9	2.4	3.8	78.2	13.3
	Mixed	0.0	1.3	1.6	28.1	26.0	43.0

Overall Performance = $\frac{\text{Number Correctly Classified}}{\text{Total Number of Samples}}$ = 72.7%

3 FEATURES BANDS 5,6,7 OCTOBER ONLY

RUNS CONFUSION MATRIX, BY PERCENTS

		Classifier Output					
Ground Truth	Open	94.7	0.0	0.5	0.0	4.8	0.0
	Marsh	20.0	12.9	8.7	27.9	25.4	5.0
	Cutover	17.1	1.3	44.7	1.3	35.5	0.0
	Conifer	0.2	1.0	1.3	93.9	2.1	1.6
	Hardwood	1.0	12.3	11.8	4.4	68.3	2.2
	Mixed	2.0	16.3	9.4	44.6	21.4	6.2

Overall Performance = 61.3%

8 FEATURES, ALL FOUR BANDS OCT. AND JAN.

		Classifier Output					
Ground Truth	Open	85.6	7.7	5.3	0.0	1.4	0.0
	Marsh	2.1	60.8	16.7	9.6	7.5	3.3
	Cutover	5.3	38.2	52.6	1.3	2.6	0.0
	Conifer	0.0	0.2	0.0	83.4	2.7	13.6
	Hardwood	0.1	3.5	3.6	3.3	76.6	12.9
	Mixed	0.0	2.4	2.0	25.0	28.0	42.6

Overall Performance = 71.3%

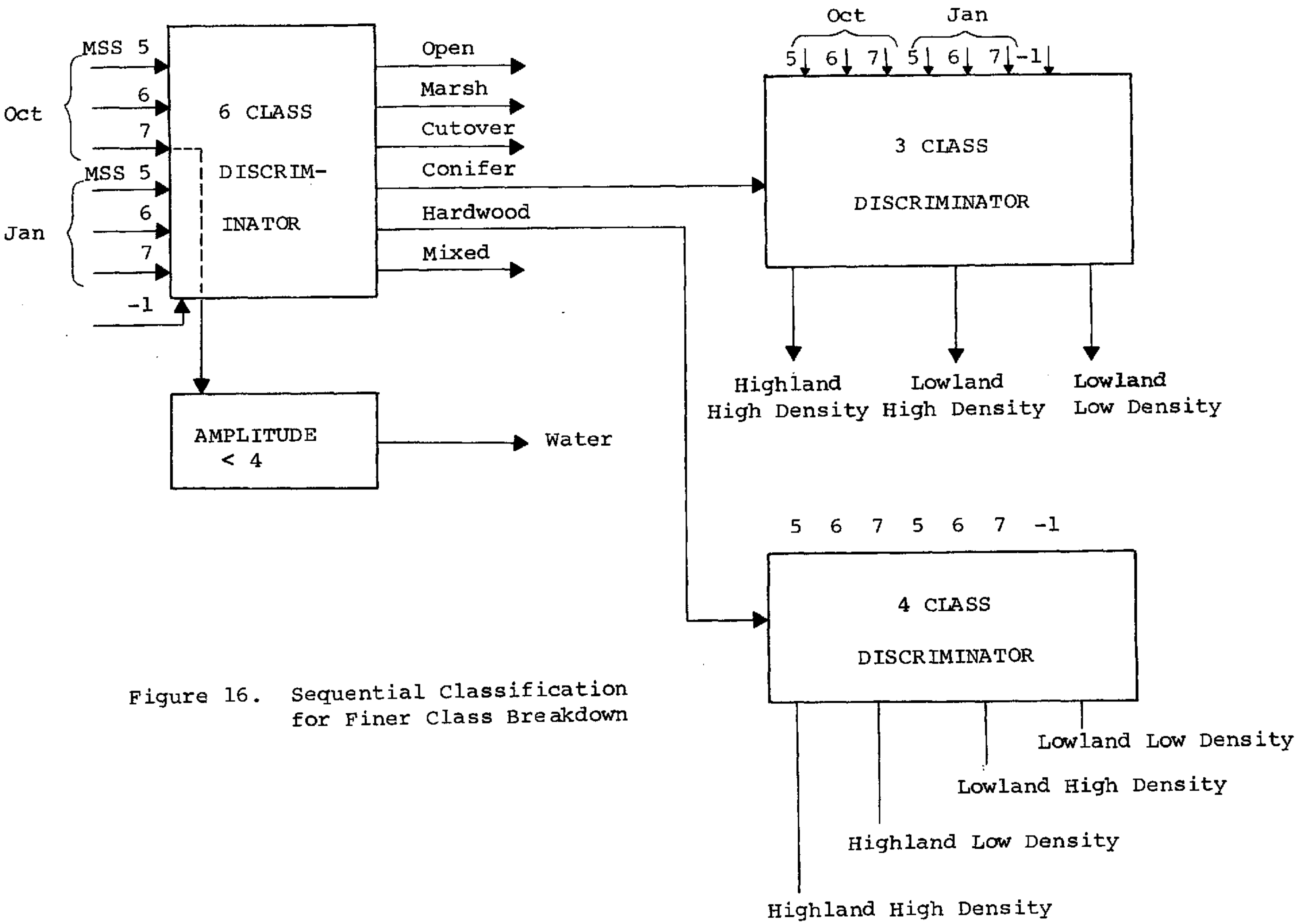


Figure 16. Sequential Classification for Finer Class Breakdown

Table VII Detailed Breakdown of Chippewa Forest

CONIFER

Ground Truth	October only (3 bands)			Oct. plus Jan. (6 bands)		
	Classifier Output			Classifier Output		
Highland/Hi Density	60.5	29.8	9.7	75.8	18.5	5.6
Lowland/Hi Density	22.0	65.3	12.7	12.3	73.5	14.2
Lowland/Low Density	25.4	23.4	51.2	9.4	15.6	75.0
	Overall performance			Overall performance		
	60.7%			74.4%		

HARDWOODS

Ground Truth	October only (3 bands)				Oct. plus Jan. (6 bands)			
	Classifier Output				Classifier Output			
Highland/Hi Density	73.0	6.5	13.7	6.9	80.9	4.9	10.5	3.7
Highland/Low Density	40.4	17.0	27.4	15.2	34.7	22.2	19.0	24.2
Lowland/Hi Density	7.9	2.6	78.9	10.5	10.5	3.9	70.4	15.1
Lowland/Low Density	10.0	16.7	50.8	22.5	10.0	10.8	17.5	61.7
	Overall performance				Overall performance			
	61.6%				69.7%			

A thematic map of the Chippewa National Forest was generated using 6 density features and sequential classification as shown in Figure 16. The classes and color assignment are listed in the following table.

Table VIII Chippewa National Forest Class Structure

Class	Symbol	Color	Pixels in West*	%	Pixels in East	%
Open and pasture	2	White	18,617	3.61	15,037	3.18
Marsh	4	Yellow Green	26,731	5.18	32,688	6.91
Cutover	5	Yellow	22,400	4.34	27,172	5.75
Highland Hi Dens. Conif.	Ah"	Brown Green	17,025	3.3	7,726	1.63
Lowland Hi Dens. Conif.	As"	Light Orange	32,669	6.33	34,080	7.21
Lowland Lo Dens. Conif.	As'	Orange	3,206	6.20	41,015	8.68
Highland Hi Dens. Hard.	Bh"	Black	131,867	25.56	128,729	27.23
Highland Lo Dens. Hard.	Bh'	Wine	14,524	2.81	19,488	4.12
Lowland Hi Dens. Hard.	Bs"	Purple	30,519	5.92	33,212	7.03
Lowland Lo Dens. Hard.	Bs'	Dark Blue	8,586	1.66	14,310	3.03
Mixed Forest	M	Green	85,454	16.56	84,148	17.80
Water	3	Blue	95,420	18.5	35,147	7.43

*To convert the number of pixels to acres, an approximate formula is

$$\text{Acreage} = 1.09 \times \text{number of pixels}.$$

A more accurate equation for computing the area of a parcel of land requires the knowledges of how many scan lines are involved in covering that parcel.

$$\text{Acreage} = 1.104 \times \text{number of pixels} + 0.453 \times \text{number of lines}$$

The thematic maps for the West and East parts of the Chippewa National Forest are shown in Figures 17 and 18. A portion of the ground truth map from the Western half west of Bowstring Lake, has been color coded with the same code as the automatic classifier output map. This is shown in Figure 19 for comparison.

The eight classes are represented by the colors shown in Table II.

Conclusions:

In order to evaluate the effectiveness of automatic classification procedures, a number of factors must be considered. Among these are the accuracy, cost, speed, type and format of the output data. Finally, but by no means of least importance, is the usefulness of the analyzed data.

Since the accuracy of automatic photointerpretation depends on a great variety of factors such as the number and distinctiveness of the classes to be delineated and the types of features utilized, one should view performance figures cautiously. In general, the classification accuracy with 5 or 6 classes ranges from 70 to 90%. Since chance performance is 16 to 20%, one can conclude that automatic interpretation is feasible. These performance figures are typical for distinguishing forest types to conifer and hardwoods. Finer distinctions introduced higher errors, for example breaking conifers into highland vs. lowland and hi density vs. low density results in an accuracy for these four classes of 69.7%. With four classes, chance would result in an accuracy of 25%, thus the improvement is not as striking.

The cost of making stratification maps automatically has to be viewed in terms of separate tasks. For example, once training weights have been obtained, one can generate thematic maps very efficiently, at a cost of about one cent for 5000 pixels. However, obtaining the training weights is a costlier operation as is the registration of the satellite with ground coordinates. A rough estimate on the cost of stratifying the State of Minnesota (54 million acres) is about $150,000 or approximately three tenths of a cent an acre. This cost would decrease rapidly for subsequent delineations as many of the tasks become repetitive and can be automated.

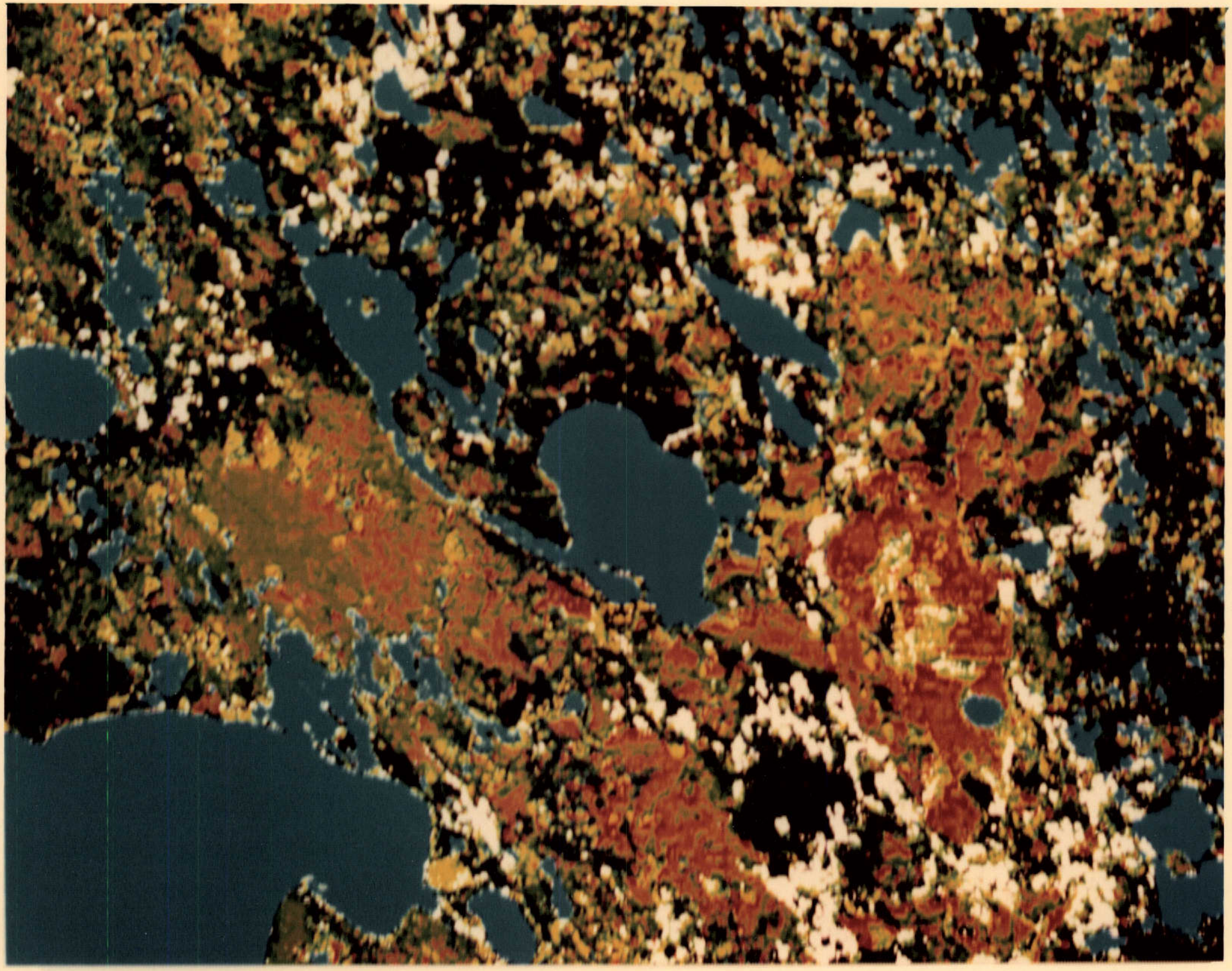


FIGURE 17. CHIPPEWA NATIONAL FOREST
THEMATIC MAP WEST, p. 49

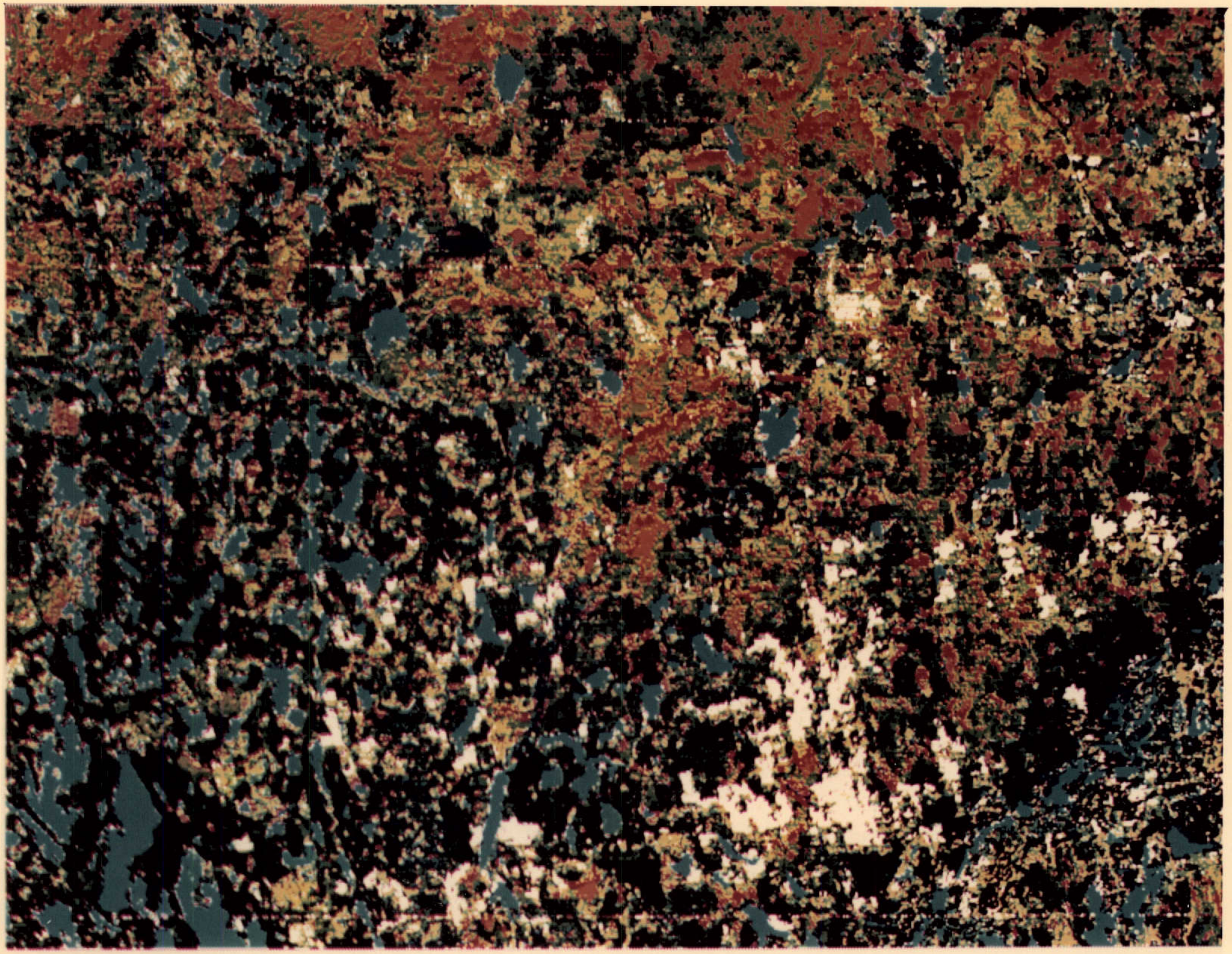


FIGURE 18. CHIPPEWA NATIONAL FOREST THEMATIC MAP EAST, p. 50

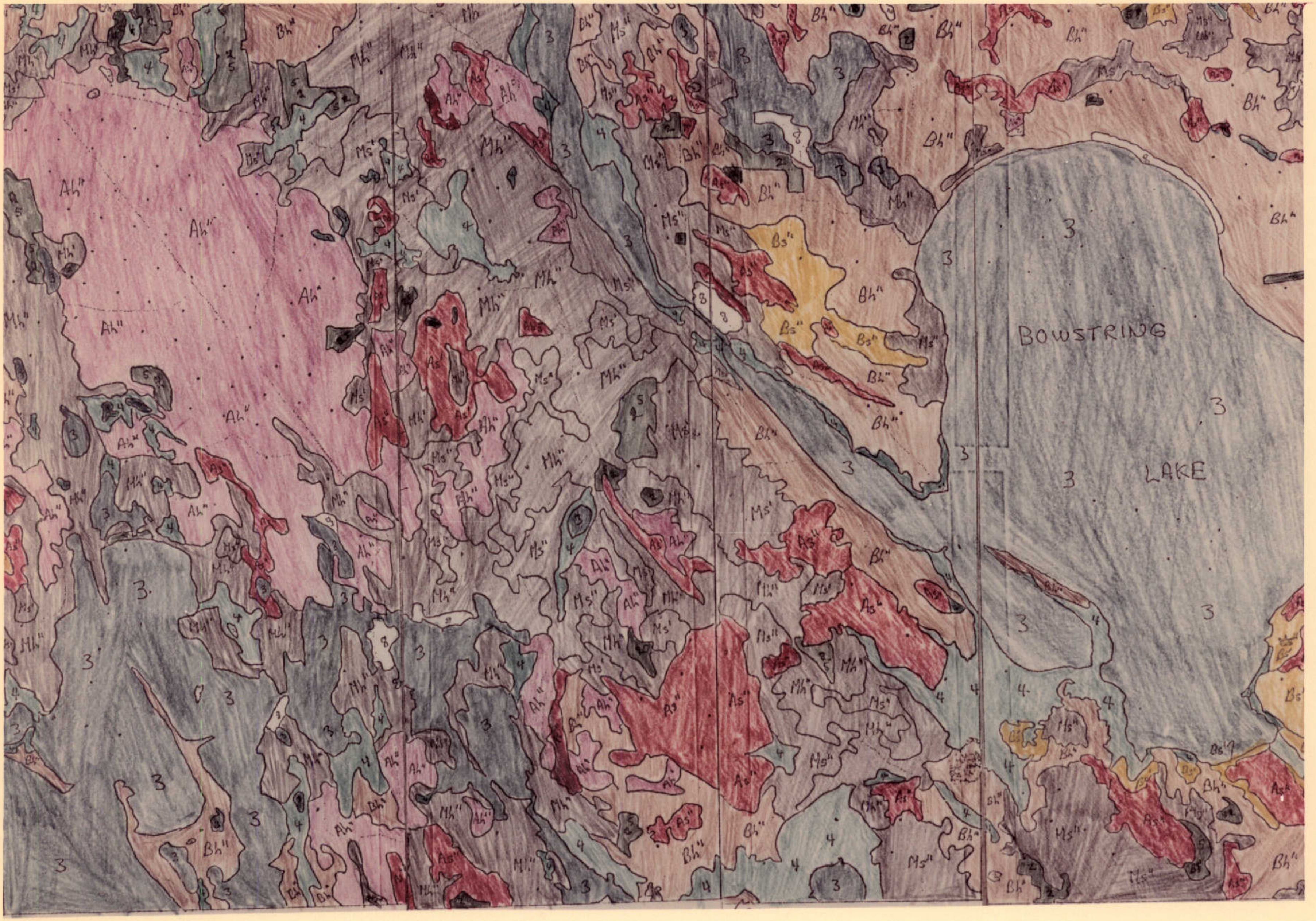


FIGURE 19. GROUND TRUTH THEMATIC MAP, CHIPPEWA WEST, p. 51

For determining the speed with which automatic classifications can be made, one must deal with the various tasks separately. Once the weights have been obtained, stratification maps can be made at a rate of 10,000 acres per minute. These figures are based on the SDS 9300 a computer with approximately 10 microseconds execution times. The time period required to obtain new weights depends on the number of classes, number of features and sample size. For five classes and 5000 training samples, one requires about 2 to 3 hours. The registration of satellite data with ground coordinates may take about the same amount of time, depending on the distortions in the data and precision of the image center coordinates.

The format of the output of an automatic classification system is ideal for a computerized inventory system. The results of such a classification scheme are consistent, repeatable and unbiased. Statistics are readily available as are color coded maps obtained on a film writer. The only human photo-interpretation required is in obtaining an accurate training set.

The use of satellite data and the automatic photo-interpretation of this data is inevitable. Its use for broad area coverage and gross information is already widely accepted. The acceptance of thematic maps delineated by automatic classifiers will require additional exposure and evaluation.

APPENDIX

Fast Fourier Transform

The Fourier Transform has been used as a tool for spectral analysis in communication systems for many years. The algorithm developed by Cooley Tukey* has made the digital implementation of the Fourier Transform feasible for a large number of sample values. Because of this Fast Fourier Transform (FFT) algorithm, one is able to tackle two dimensional image problems.

A flow chart of procedures for computing the FFT algorithm in one dimension is shown in Figure 20. Two dimensionality is obtained by applying the algorithm first to rows and then to columns as shown in figure 21.

Since the signal input is an 8 x 8 array of real numbers, the complex part of the input to the algorithm is initially set to zero. Upon applying the algorithm, one obtains two sets of 8 x 8 arrays comprising the real and imaginary part of the output.

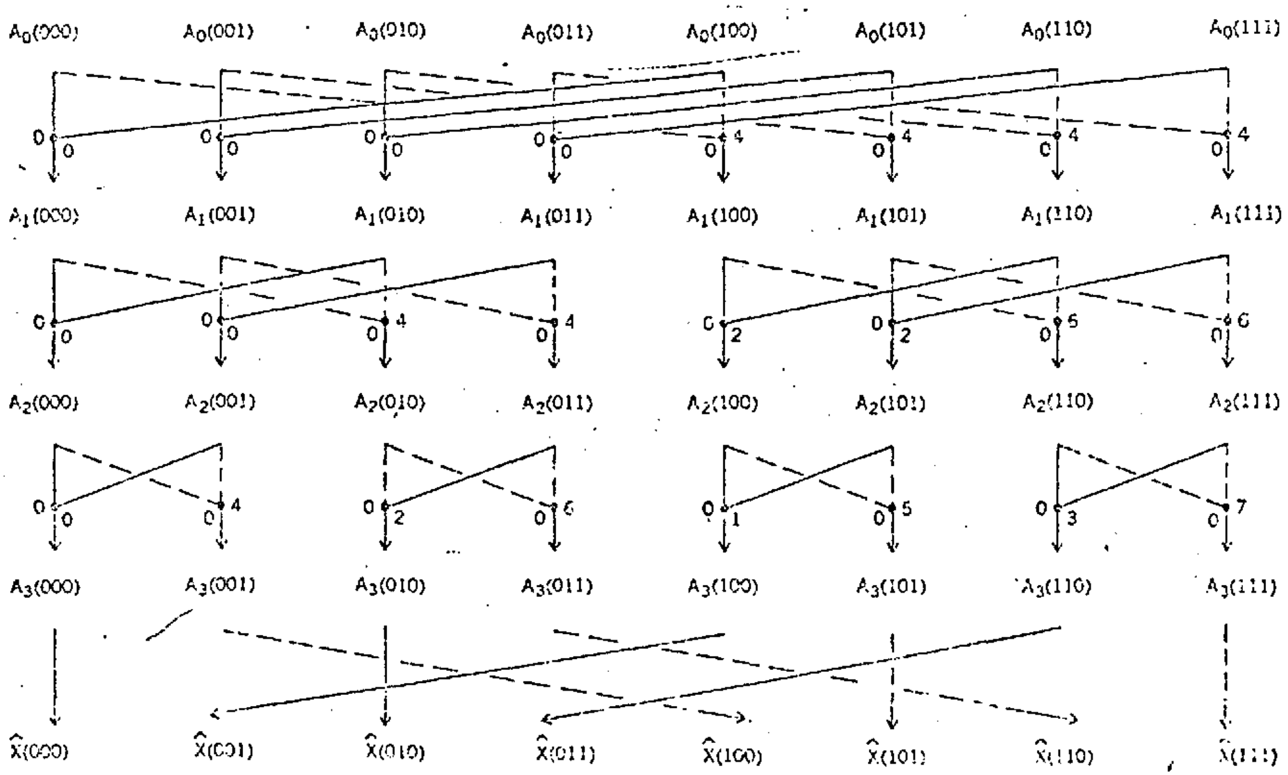
The Cooley-Tukey Algorithm is then applied to the columns of the signal obtained from the first pass. The input consists of both a real and imaginary 8 x 8 matrix. The output is unscrambled by a bit reversing technique which is applied to rows and then to columns. The unscrambling amounts to a bit reversal as shown in the bottom line of Figure 20. The power matrix is obtained by taking the sum of the squares of the real and imaginary terms.

The first four terms of the output power matrix represent the Fourier Coefficients of the first four harmonics. The fifth term represents only one of the two fifth harmonic sinusoidal components. The last three terms are repetitions of term 1,2, and 3.

*Cooley, J. W. and Tukey, J. W., "An Algorithm for Machine Calculation of Complex Fourier Series", Mathematics of Computation, 19, 297-301 (April 1965).

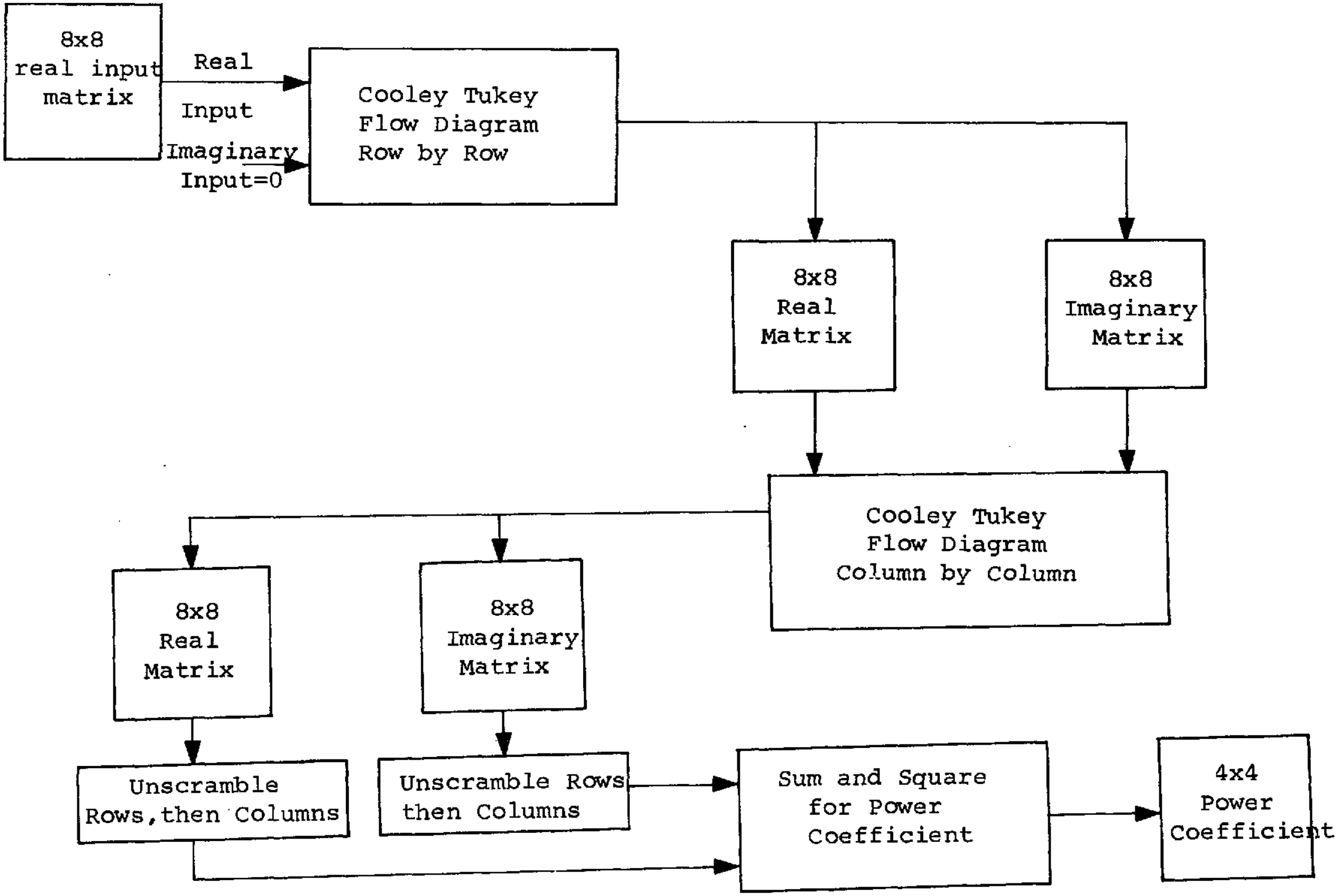
When 25 spectral components are used for features as shown on Figures 11 and 12, the first 5 x 5 terms are used. For 16 features, the first 4 x 4 terms are used.

Figure 20. A flow diagram of the Cooley-Tukey FFT Algorithm for Performing an Eight-Point Transform.



The power coefficients from the two MSS bands can be computed simultaneously by using one band as the real input and the second band as the imaginary input to the Cooley-Tukey flow diagram. By appropriate unscrambling, the power coefficients can be determined for both input signals.

FIGURE 21 2-DIMENSIONAL COOLEY TUKEY ALGORITHM



Fast Walsh Transform

The Fast Walsh Transform* is a texture measure which uses square waves for basis vectors in contrast to sinusoids used in the Fourier transform. The square wave (one minus one) notation makes the transform particularly amenable to digital computers.

The required procedures for the computation of the Walsh coefficients are shown in Figure 22. For an 8 x 8 matrix, the Hadamard matrix elements are the following:

$$\begin{matrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1
\end{matrix}$$

The computational method used is identical to the flow chart which is shown forthe FFT if one replaces the Cooley-Tukey Flow Diagram by the Hadamard Flow Diagram shown in Figure 22.

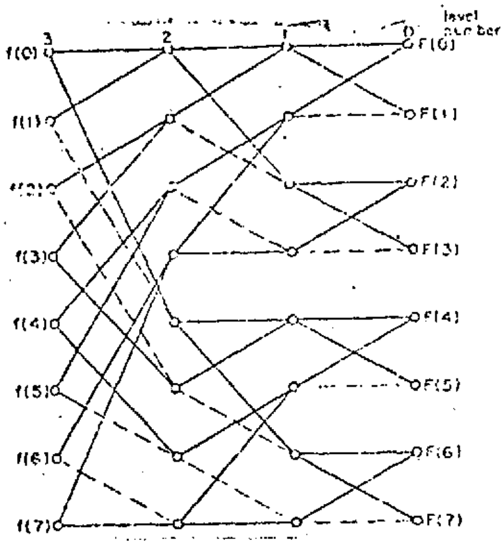


Figure 22. A flow diagram of the Fast Walsh Algorithm for performing an eight-point transform.

*Pratt, W. K. et al., "Transform Image Coding", NASA-CR-110153 Univ. of So. California, March, 1970, pp. 154.

Figure 22 illustrates the computations performed for a one-dimensional Hadamard transformation with eight data points. The data points are arranged in a column at level three and then summed by pairs to produce intermediate results for level two. A dotted line linking two modes indicates that the data point at the higher level is multipled by minus one before addition, or equivalently, the data point forms the subtrahend of a subtraction operation. Operations follow the tree graph to level 0 which is the ordered Hadamard transform of f(x). There are two operations performed at each node of levels 0, 1 and 2 yielding a total of eight log 8 = 24 operations.

Similar to the FFT, the two dimensional FWT transform is obtained by first computing the coefficients by row and then repeating the calculation by column. The Hadamard transform coefficients are then squared and summed to obtain the five by five power coefficient matrix.

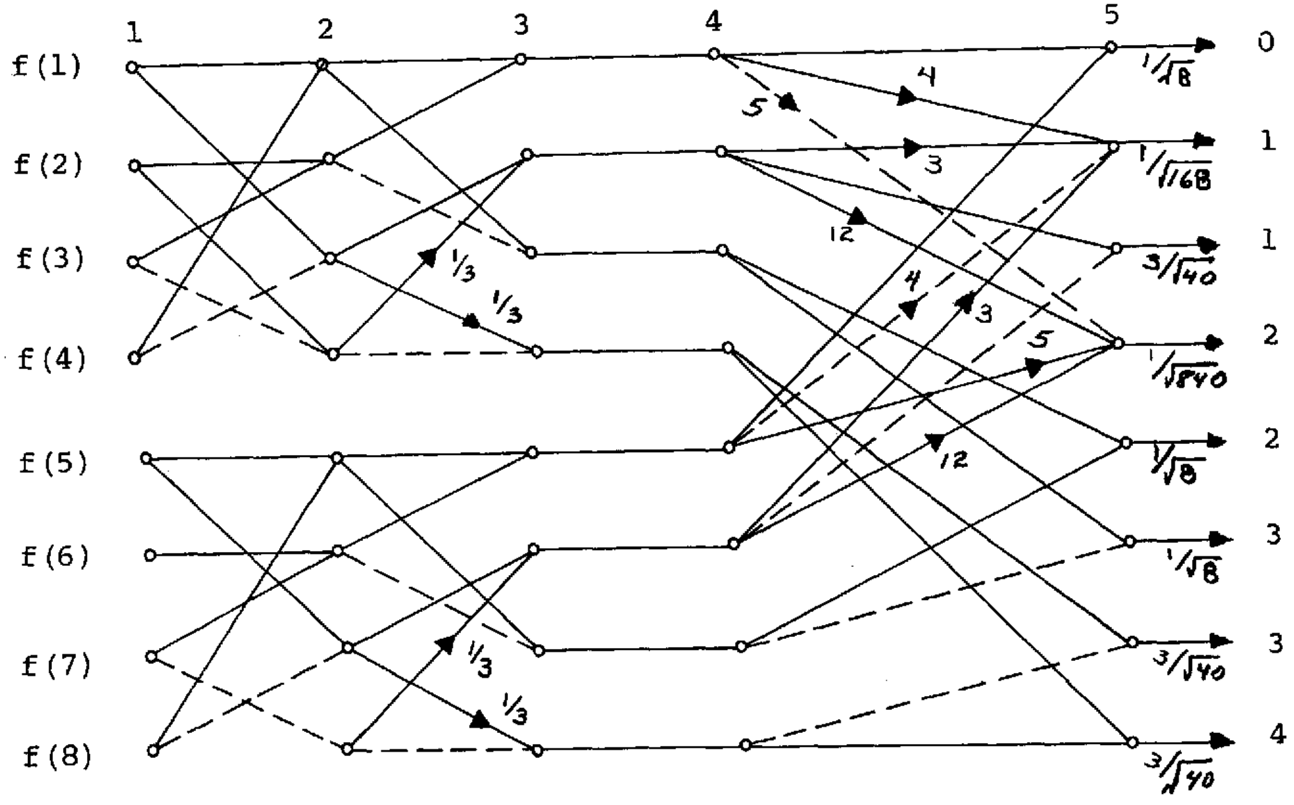
Slant Transform

The slant transform is a new orthogonal image transform* with a basis vector matched to the gradual brightness changes along image lines. This transform can be computed using a fast computational algorithm. The computational flow chart for the slant transform of order eight is shown in Figure 23. The flow chart also includes the reordering of terms.

A desirable property for an image coding transform is that the transform compact the image energy to as few of the transform domain samples as possible. A high degree of energy compaction will result if the basis vectors of the transform matrix resemble horizontal or vertical lines of an image. The slant vector is a discrete sawtooth waveform decreasing in uniform steps over its length and thus is suitable for efficiently representing gradual brightness changes in an image line.

*Pratt, W.K. et.al, "Slant Transform for Image Coding" 1972 Proceedings Applications of Walsh Functions, March 1972 pp. 229-234. This reference is included at the end of this report. A listing for computing an 8x8 set of coefficients follows this reference.

FIGURE 23 FLOW CHART FOR COMPUTING AN EIGHTH ORDER SLANT TRANSFORM



Karhunen Loeve Transform

The objective of the Karhunen Loeve expansion (or Principal Components Analysis) is to find a lower-dimensional representation based on the variance of the features. This is achieved by a linear transformation described by an orthogonal matrix. This transformation is equivalent to a rotation of the original pattern space to a new set of coordinate vectors which are also orthogonal and provide a means for dimensionality reduction.

For example, we can take the four MSS spectral bands and regard these as a vector. If the number of vectors in the training set is N, the set of N vectors constitutes a matrix of order 4 x N. Let us call this matrix Z. Let Z' be the matrix transpose of Z. The matrix Z Z' is a 4 x 4 square matrix. The eigenvectors of this matrix represent the preferred set of basis multispectral bands. If these eigenvectors are arranged in the order of decreasing magnitude of their associated eigenvalues, the vectors have been ordered according to their importance. If only the first k of these eigenvectors are used ($k < 4$), the residual variance is less than for any other set of k vectors chosen by any other means. In other words, this is the optimum expansion based on the data set itself. The eignevectors are functions defined by the domain consisting of the 4 points in the local array.

One can consider the 64 points of an 8 x 8 array from one MSS band as the components of a vector. If these 64 points composing the local image are ordered in a consistent way, the intensity at these 64 points will form the vector. If one considers N training samples of this vector, the set of N vectors forms a matrix of order 64 x N. From this matrix one can compute the covariance matrix. The covariance matrix can be analyzed by the method of principal components. The eigenvectors of this matrix provide a preferred set of basis elements for representing the local images. If one considers these eigenvectors in the order of decreasing magnitude of their associated eigenvalues, they will be ordered according to their importance. This procedure was used to select the most important 9, 16 or 25 features as shown in Figure 12.

The Karhunen Loeve Transform provides the optimum compaction of image energy but requires a great deal of computation. First, the image covariance matrix must be computed. Then the covariance matrix must be diagonalized to determine its eigenvalues and eigenvectors. There is no fast algorithm for this transform.

A more detailed description of this algorithm can be found in a number of references.*

1. P. J. Ready and P. A. Wintz, IEEE Trans. on Comm., Vol. COM-21, #10, 1123 (1973).

2. J. M. Mendel and K. S. Fu, Adaptive, Learning and Pattern Recognition Systems, Academic Press, New York (1970).

3. J. Spragins, IEEE Trans. Inf. Theory IT-12, 223 (1966).

4. Ya Z. Tsypkin, Fundamentals of Learning Systems Theory, Nauka, Moscow (1970), In Russian.

5. J. Kittler and P. C. Young, Cambridge University Technical Report, CUED B-Control/TR29 (1972).

6. Y. T. Chien and K. S. Fu, IEEE Trans. Inf. Theory IT-15, 518 (1967).

7. S. Watanabe et al., Computers and Information Sciences, Vol. II. Academic Press, New York (1967).

8. S. Watanabe, Proc. 4th Prague Conf. on Information Theory (1965).

9. J. Kittler and P. C. Young, Pattern Recognition, Vol. 5, 335 (1973).

Acknowledgements

The author acknowledges with thanks the valuable assistance of Mr. Michael Cheung and Mr. Mohammed Ennouri in the computer programming and data processing essential to implementation of the ideas presented in this report. The consultation provided by Mr. Ferdinand Ohnsorg has been indispensable in the implementation of the various **texture** algorithms used in this study. For supplying the ground truth data the author acknowledges with thanks the arduous work of Mr. Gregg R. Johnson and Mark Jensen of the University of Minnesota School of Forestry. The kind cooperation, encouragement and user contacts provided by Dr. Merle P. Meyer and Dr. Joseph Ulliman, professors at the School of Forestry, are also gratefully acknowledged.

Finally, but of primary importance, were the suggestions, coordination and support provided by Mr. George D. Swanlund, which made this study possible.

SLANT TRANSFORMS FOR IMAGE CODING*

William K. Pratt
Lloyd R. Welch
Wen-hsiung Chen
Department of Electrical Engineering
University of Southern California
Los Angeles, California

Introduction

In 1968 the concept of coding and transmitting the two dimensional Fourier transform of an image, computed by a fast computational algorithm, rather than the image itself, was introduced [1,2]. This was followed shortly thereafter by the discovery that the Walsh-Hadamard transform could be utilized in place of the Fourier transform with a considerable decrease in computational requirements [3]. Investigations then began into the application of the Karhunen-Loeve [4] and the Haar [5] transforms for image coding. The Karhunen-Loeve transform provides minimum mean square error coding performance but does not possess a fast computational algorithm. On the other hand the Haar transform has the attribute of an extremely efficient computational algorithm, but results in a relatively large coding error. None of the transforms mentioned above, however, has been expressly tailored to the characteristics of an image.

A desirable property for an image coding transform is that the transform compact the image energy to as few of the transform domain samples as possible. Qualitatively speaking, a high degree of energy compaction will result if the basis vectors of the transformation matrix "resemble" typical horizontal or vertical lines of an image. If one examines the lines of a typical monochrome image, it is found that a large number of the lines are of nearly constant grey level over a considerable length. The Fourier, Hadamard, and Haar transforms possess a constant valued basis vector that provides an efficient representation for constant grey level image lines, while the Karhunen-Loeve transform has a nearly constant basis vector suitable for this representation. Another type of typical image line is the line that linearly increases or decreases in brightness over the length. None of the data transforms previously mentioned possess a basis vector that efficiently represents such image lines.

Shibata and Enomoto have introduced orthogonal transformations containing a "slant" basis vector for data of vector lengths of four and eight [6]. The slant vector is a discrete sawtooth waveform decreasing in uniform steps over its length, and is suitable for efficiently representing gradual brightness changes in an image line. Their work gives no indication of a construction for larger size data vectors, nor exhibits the use of a fast computational algorithm. In order to achieve a high degree of image coding compression with transform coding techniques, it is necessary to perform the transformation in two dimensions over block sizes of 16 x 16 picture elements or greater [7]. For large block sizes, computation is usually not feasible unless a fast algorithm is employed.

With this background an investigation was undertaken to develop an image coding slant transform matrix possessing the following properties:

1. orthogonal set of basis vectors.
2. constant basis vector.
3. slant basis vectors.
4. sequency property.
5. variable size transformation.
6. fast computational algorithm.
7. high energy compaction.

The following sections describe the construction of the slant transformation matrix, present a fast computational algorithm for its computation, discuss its image coding performance, and provide examples of its use for coding monochrome and color images.

Slant Transform Construction

For a vector length of N = 2 the slant transform is identical to the Hadamard transform of order 2. Thus,

$$[S_2] = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad (1)$$

*This work was supported by the Advanced Research Projects Agency of the Defense and was monitored by the Air Force Eastern Test Range under Contract No. F08606-72-C-0008.

The slant transform matrix for N = 4 can be written as

$$[S_4] = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ a+b & a-b & -a+b & -a-b \\ 1 & -1 & -1 & 1 \\ a-b & -a-b & a+b & -a+b \end{bmatrix} \qquad (2)$$

where a and b are real constants to be determined subject to the conditions that S_4 must be orthogonal and that the step size of the slant basis vector must be the same throughout its length. The step size between the first two elements of the slant vector is

$$(a+b) - (a-b) = 2b \qquad (3)$$

and the step size between the second and third elements is

$$(a-b) - (-a+b) = 2a - 2b \qquad (4)$$

Hence,

$$a = 2b$$

The slant matrix of order four may then be reformed as

$$[S_4] = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3b & b & -b & -3b \\ 1 & -1 & -1 & 1 \\ b & -3b & 3b & -b \end{bmatrix} \qquad (5)$$

By the orthogonality condition

$$\frac{1}{\sqrt{4}} [3b \;\; b \;\; -b \;\; -3b] \frac{1}{\sqrt{4}} [3b \;\; b \;\; -b \;\; 3b]^T = 1$$

it is found that

$$b = \frac{1}{\sqrt{5}} \qquad a = \frac{2}{\sqrt{5}}$$

Thus, the slant matrix becomes

$$[S_4] = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} \\ 1 & -1 & -1 & 1 \\ \frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} \end{bmatrix} \qquad (6)$$

It is easily shown that S_4 is orthonormal. Further note that S_4 possesses the sequency property; each row has an increasing number of sign reversals from 0 to 3. The fast computational property of S_4 is apparent from the matrix decomposition

$$[S_4] = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{3}{\sqrt{5}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{3}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{3} \\ 1 & -1 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \qquad (7)$$

If S_4 is post multiplied by a column data vector, the first computational pass requires 4 additions, the second pass requires 2 multiplications (the two elements 1/3) and the final pass requires 4 multiplications including the normalizing factor of $1/\sqrt{4}$. The total computational requirements are 8 adds and 6 multiples. For purposes of comparison a fourth order Hadamard transform requires 8 adds and 4 multiples. Figure 1 contains a flow chart of the computational operations for S_4.

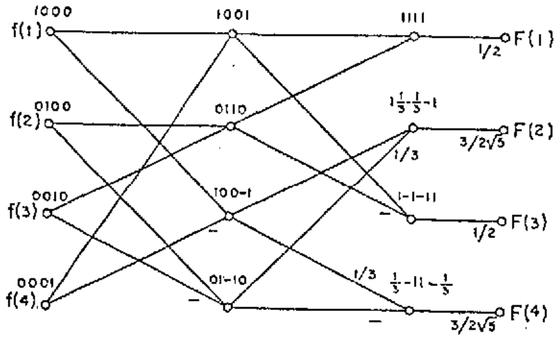


Figure 1. Slant transform of order 4-computational flowchart.

An extension of the slant matrix to its next size increment S_8 is given by

$$[S_8] = \frac{1}{\sqrt{8}} \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ a_8 & b_8 & 0 & 0 & -a_8 & b_8 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ -b_8 & a_8 & 0 & 0 & b_8 & a_8 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{array}\right] \left[\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & & & & \\ \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & & & 0 & \\ 1 & -1 & -1 & 1 & & & & \\ \frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & & & & \\ \hline & & & & 1 & 1 & 1 & 1 \\ & 0 & & & \frac{3}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{-1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} \\ & & & & 1 & -1 & -1 & 1 \\ & & & & \frac{1}{\sqrt{5}} & \frac{-3}{\sqrt{5}} & \frac{3}{\sqrt{5}} & \frac{-1}{\sqrt{5}} \end{array}\right] \qquad (8)$$

where a_8 and b_8 are constants to be determined to satisfy the slant and sequency properties. In S_8 the slant vector is obtained by a simple scaling operation on S_4. The remaining terms in eq. (8) are introduced to obtain the sequency and orthogonality properties.

Equation (8) can be generalized to give the slant matrix of order N in terms of the slant matrix of order N/2 by the following construction.

$$[S_N] = \frac{1}{\sqrt{2}} \left[\begin{array}{cc|c|cc|c} 1 & 0 & 0 & 1 & 0 & 0 \\ a_N & b_N & & -a_N & b_N & \\ \hline & & I & & & I \\ 0 & & & 0 & & \\ & & & & & \\ \hline 0 & 1 & 0 & 0 & -1 & 0 \\ -b_N & a_N & & b_N & a_N & \\ \hline & & I & & & -I \\ 0 & & & 0 & & \end{array}\right] \left[\begin{array}{c|c} [S_{N/2}] & 0 \\ \hline 0 & [S_{N/2}] \end{array}\right] \quad (9)$$

where I represents a 2 X 2 identity matrix. To determine the coefficients (a_N, b_N), one proceeds as follows: The first row is a constant

$$S_N(1,i) = \frac{1}{\sqrt{N}}$$

The second row (the slant vector) is a linear function of the column index which is orthogonal to the first row. It must therefore be of the form

$$S_N(2,i) = x_N(N+1-2i)$$

Now, by the recursion indicated in eq. (9), for $i \leq N$

$$S_{2N}(2,i) = \frac{1}{\sqrt{2}} a_{2N} S_N(1,i) + \frac{1}{\sqrt{2}} b_{2N} S_N(2,i)$$

or

$$x_{2N}(2N+1-2i) = \frac{1}{\sqrt{2N}} a_{2N} + b_{2N} \frac{x_N}{\sqrt{2}} (N+1-2i)$$

From this one obtains

$$x_{2N} = b_{2N} \frac{x_N}{\sqrt{2}}$$

$$a_{2N} = \sqrt{2}\, N^{3/2} x_{2N}$$

and by induction

$$a_{2N} = 2b_{2N} a_N$$

Since $S_N(1,\cdot)$ and $S_N(2,\cdot)$ are orthogonal unit vectors in N dimensions and $S_{2N}(2,\cdot)$ is a unit vector in 2N dimensions, the above recursion implies

$$1 = \|S_{2N}(2,\cdot)\| = a_{2N}^2 + b_{2N}^2$$

These two relations can be used to obtain the coefficients, (a_N, b_N) recursively:

$$a_1 = 1$$

$$b_{2N} = 1/\sqrt{1 + 4a_N^2}$$

$$a_{2N} = 2b_{2N} a_N$$

Figure 2 contains a superimposed plot of the Walsh-Hadamard and Slant basis vectors for a vector length of sixteen for the construction of eq. (9). It is interesting to note that many of the mid-sequency basis vectors are identical for the two transforms.

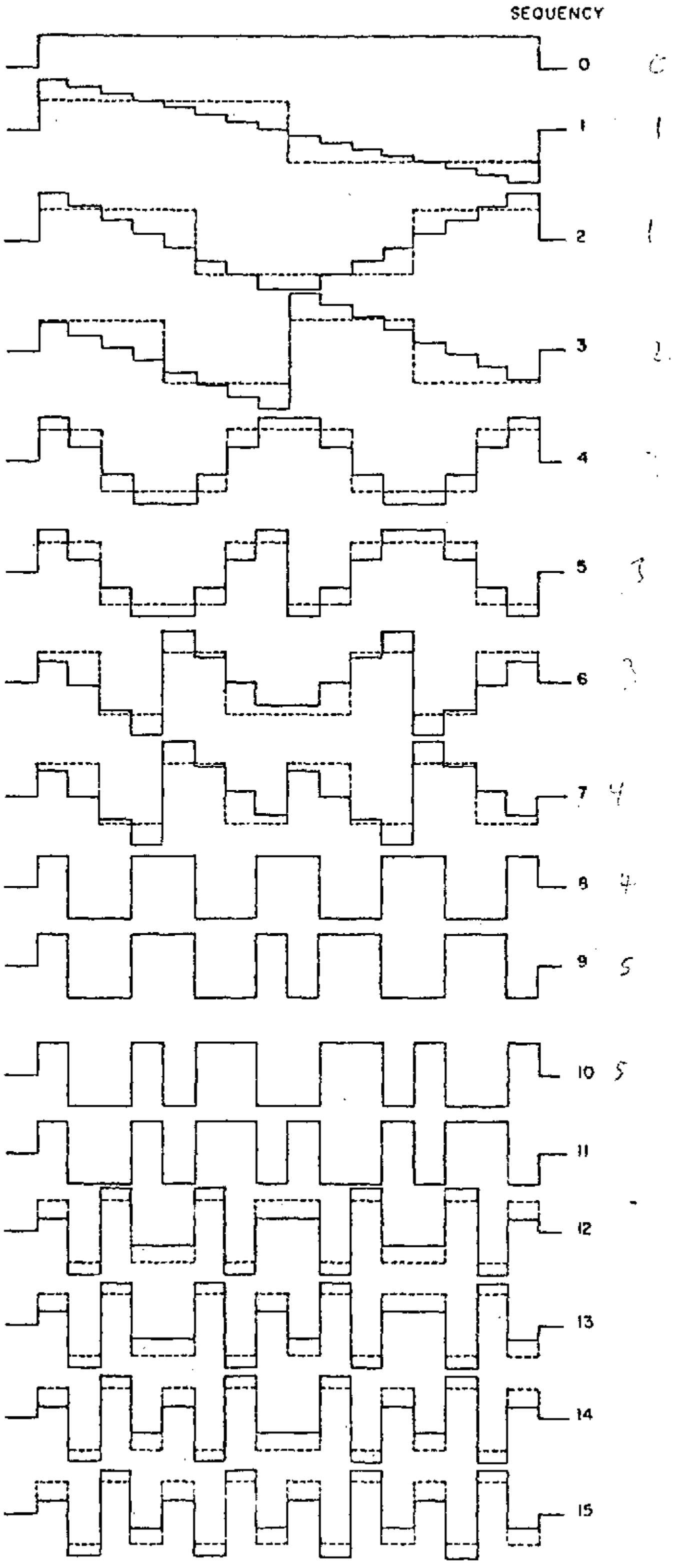


Figure 2. Comparison of Walsh-Hadamard and Slant basis vectors of length 16.

Slant Transform Properties

Let $[f]$ be a column data vector and $[F]$ its slant transform obtained by the operation

$$[F] = [S_N][f] \tag{10}$$

Consider $[f]$ to be a sample of a vector random process with known mean $[\bar{f}]$ and with a known covariance matrix

$$[C_f] = \overline{([f] - [\bar{f}])([f]^* - [\bar{f}]^*)} \tag{11}$$

where the overbar indicates a statistical average. The covariance matrix of the Slant transform $[F]$ is found to be [8]

$$[C_F] = [S_N][C_f][S_N]^T \tag{12}$$

Equation (12) can be considered a two dimensional Slant transformation of the data covariance matrix for purposes of computation. Figure 3 contains a perspective view of the Slant transform of a data vector of length N = 256 with a Markov process covariance of the form

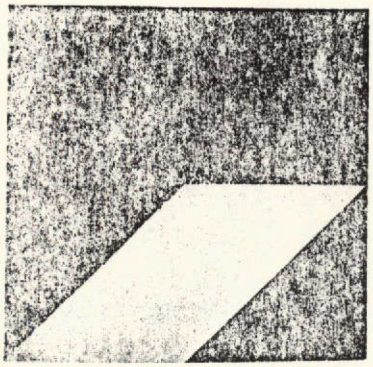


Figure 3. Perspective view of Slant transform covariance matrix-Markov process data vector, $\rho = 0.95$, N = 256.

$$[C_f] = \rho^{|x_i - x_j|}$$

where ρ is the correlation of adjacent elements $[f]$. Figure 4 is a plot of the variance of the Slant transform samples as a function of sequency. The variance functions for the Walsh-Hadamard, Fourier, Haar, and Karhunen-Loeve transforms are included for comparison. It is seen that the variance function for the Slant transform is reasonably close to the variance function of the Karhunen-Loeve transform, which is known to provide the best energy compaction for the Markov source.

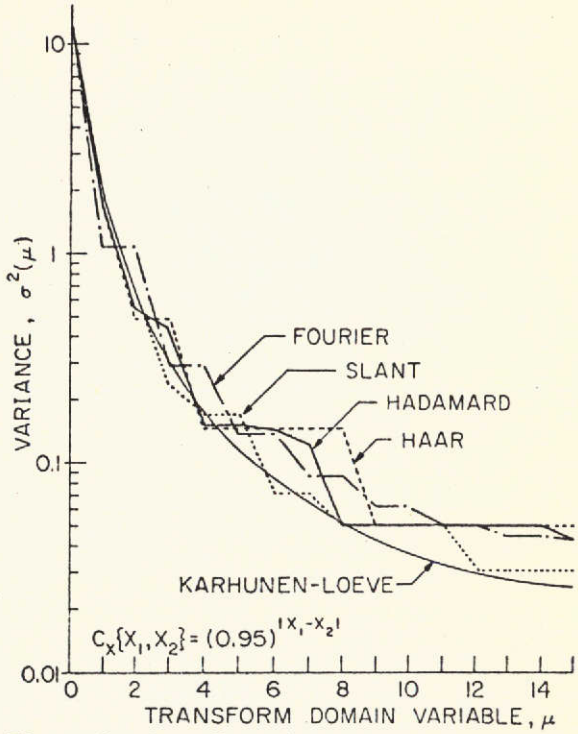


Figure 4. Transform domain variance, vector length = 16, element correlation = 0.95.

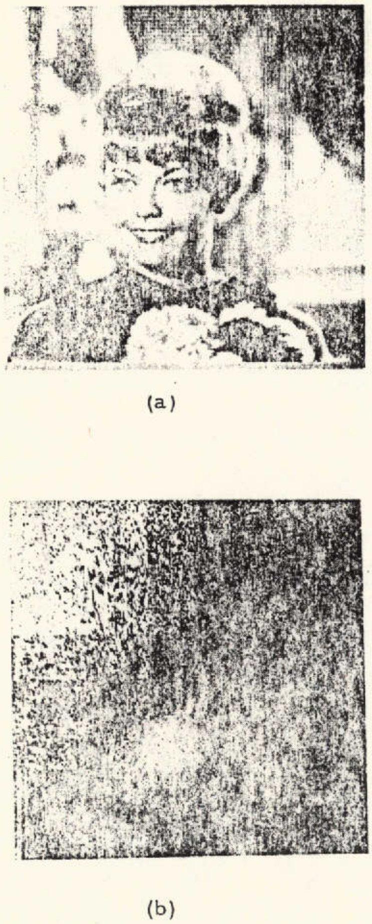
Slant Transform Image Coding

Let $[f(x,y)]$ represent the brightness samples of an N by N element image. The two dimensional Slant transform of the image is given by

$$[F(u,v)] = [S_N][f(x,y)][S_N]^T$$

In effect, the pre-multiplication of $[f(x,y)]$ by $[S_N]$ performs a one dimensional slant transform of each column of the image matrix, and the post-multiplication by $[S_N]^T$ performs a one dimensional transform of the rows of the image. Figure 5 contains a photograph of a 256 by 256 element image with 64 grey levels and its two dimensional Slant transform.

A bandwidth reduction can be obtained with the Slant transform by efficiently quantizing each transform domain sample. There are two basic strategies for the quantization process - zonal and threshold quantization. In the former, various zones are established in the transform domain, and each sample in the zone is coded with the same number of bits set proportional to the expected variance of the samples within the zone. With threshold quantization a threshold level is established and only those transform domain samples whose magnitude are greater than the threshold are coded.


(a)

(b)

Figure 5. Slant transform of an image:(a) original; (b) transform threshold display.

Figure 6 presents a statistical evaluation of the coding performance of the Walsh-Hadamard, Karhunen-Loeve, and Slant transforms for a form of zonal quantization in which the transform domain samples in a zone are coded with six bits per sample and samples outside the zone are discarded. The zone is defined to contain the transform domain samples with the largest expected variance, and is adjusted to include 25% of the total number of transform domain samples. Images coded with this system require an average coding of 1.5 bits per element. Figure 6 plots the mean square error resulting from this quantization process as a function of the size of the image block transformed. From the figure it is seen that the Karhunen-Loeve transform provides the minimum mean square error, but the Slant transform results in only a slightly greater error. Also to be noted is that the rate of decrease in mean square error for larger block sizes becomes quite small after a block size of 32 by 32 elements.

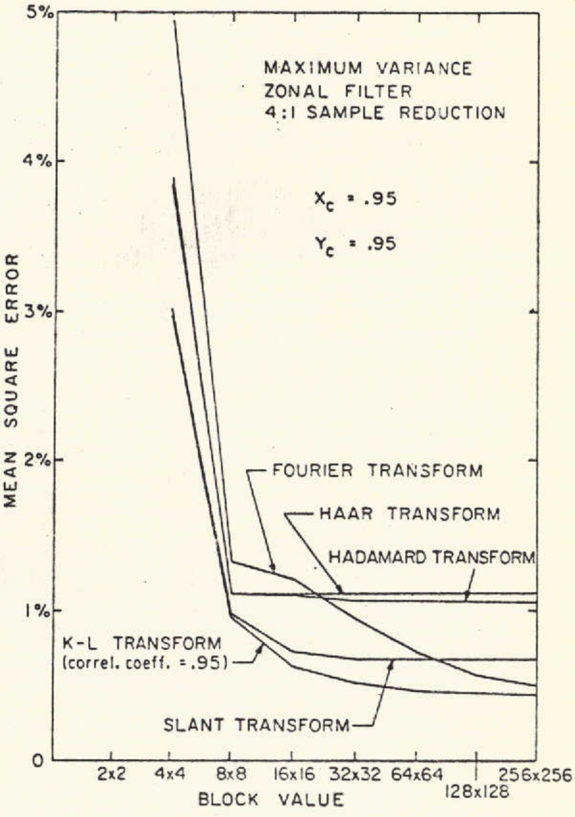
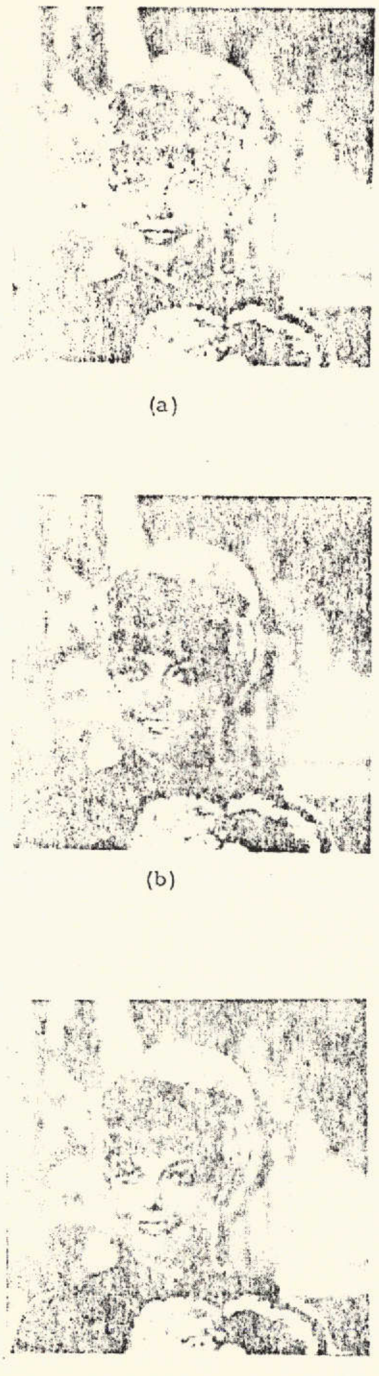


Figure 6. Mean square error performance of image transforms as a function of block size for low pass zonal quantization.

Several computer simulations have been performed to evaluate the Slant transform for image coding. Figure 7 shows image reconstructions for the Walsh-Hadamard, Karhunen-Loeve, and Slant transforms for zonal quantization employing eight zones and coding with an average of only 1.5 bits per element. Subjectively, the Slant transform results in much less degradation than the Walsh-Hadamard transform and only slightly more than the Karhunen-Loeve transform. Similar experiments have been performed for color images, and it has been found that a color image can be coded with about 2.0 bits per element with negligible degradation using the Slant transform.



(a)

(b)

(c)

Figure 7. Reduced bandwidth images obtained by transform coding with zonal quantization, 1.5 bits/pixel. (a) Walsh-Hadamard coded; (b) Slant coded; (c) Karhunen-Loeve coded.

Summary

A new orthogonal image transform with a basis vector matched to gradual brightness changes along image lines has been developed. The transform can be computed using a fast computational algorithm, and requires only a few more operations than the Walsh-Hadamard transform. A statistical analysis indicates that the Slant transform provides a smaller mean square error for image coding than the Walsh-Hadamard transform and a slightly greater error than the Karhunen-Loeve transform. The analytic image coding performance predictions are verified by computer simulations of image coding processes for monochrome and color images.

References

1. H. C. Andrews and W. K. Pratt, "Television Bandwidth Reduction by Encoding Spatial Frequencies," Journal of the SMPTE, Vol. 77, No. 2, December, 1968, pp. 1279-1281.

2. G. B. Anderson and T. S. Huang, "Piecewise Fourier Transformation for Picture Bandwidth Compression," IEEE Transactions on Communication Technology, Vol. COM-19, No. 2, April, 1971, pp. 133-140.

3. H. C. Andrews, J. Kane, and W. K. Pratt, "Hadamard Transform Image Coding," Proceedings of the IEEE, Vol. 57, No. 1, January, 1969.

4. A. Habibi and P. A. Wintz, "Image Coding by Linear Transformation and Block Quantization," IEEE Transactions on Communication Technology, Vol. COM-19, No. 1, February, 1971, pp. 50-63.

5. H. C. Andrews, Computer Techniques in Image Processing, Academic Press, New York, 1970.

6. H. Enomoto and K. Shibata, "Orthogonal Transform Coding System for Television Signals," Journal of the Institute of TV Engineers of Japan, Vol. 24, No. 2, February, 1970, pp. 99-108.

7. W. K. Pratt, "A Comparison of Digital Image Transforms," University of Missouri at Rolla, Communications Conference, September, 1970.

8. W. K. Pratt, "Linear and Nonlinear Filtering in the Walsh Domain," Proceedings of the Symposium on the Application of Walsh Functions, April, 1971, pp. 38-42.

```
*JOB KOOLY
*ASSIGN X1=NO
*ASSIGN 3=CR1A,1=LP1B
*FORTRAN LS,GO
       COMMON F(8),FP(8),M(4),A(8,8)
       DIMENSION AI(8),AJ(8)
       READ(3,10) AI,AJ
       DO 5 I=1,8
       DO 5 J=1,8
     5 A(I,J)=SQRT(2.)*AI(I)*AJ(J)/10.
    10 FORMAT(8F5.0)
       WRITE(1,20) ((A(I,J),J=1,8),I=1,8)
    20 FORMAT(1H0,(8F10.5))
       DO 40 J=1,8
       DO 30 I=1,8
    30 F(I)=A(I,J)
       CALL SLNTX
       DO 40 I=1,8
    40 A(I,J)=FP(I)
       WRITE(1,20)  ((A(I,J),J=1,8),I=1,8)
       DO 60 I=1,8
       DO 50 J=1,8
    50 F(J)=A(I,J)
       CALL SLNTX
       DO 60 J=1,8
    60 A(I,J)=FP(J)
       DO 70 I=1,8
       DO 70 J=1,8
    70 A(I,J)=A(I,J)/8.
       WRITE(1,20)  ((A(I,J),J=1,8),I=1,8)
       STOP
       END
       SUBROUTINE SLNTX
       COMMON F(8),FP(8),M(4),A(8,8)
       DATA C1,C2,C3/.2182178902,1.341640787,.0975900073/
       DO 10 J=1,2
       K=4*(J-1)+1
       FP(K)=F(K)+F(K+3)
       FP(K+1)=F(K+1)+F(K+2)
       FP(K+2)=F(K)-F(K+3)
       FP(K+3)=F(K+1)-F(K+2)
       F(K)=FP(K)+FP(K+1)
       F(K+1)=FP(K+2)+FP(K+3)/3.0
       F(K+2)=FP(K)-FP(K+1)
       F(K+3)=FP(K+2)/3.0-FP(K+3)
    10 CONTINUE
       FP(1)=F(1)+F(5)
       FP(2)=C1*(4.0*(F(1)-F(5))+3.0*(F(2)+F(6)))
       FP(3)=(F(2)-F(6))*C2
       FP(4)=C3*(12.0*(F(2)+F(6))+5.0*(F(5)-F(1)))
       FP(5)=F(3)+F(7)
       FP(6)=F(3)-F(7)
       FP(7)=C2*(F(4)-F(8))
       FP(8)=C2*(F(4)+F(8))
       RETURN
       END
*LOAD XM
*DATA
    3    1   -1   -3   -3   -1    1    3
    7   -1   -9  -17   17    9    1   -7
*FIN
```